AN ETHNOGRAPHICALLY-INFORMED ANALYSIS OF THE INFLUENCE OF CULTURE ON GLOBAL SOFTWARE-TESTING PRACTICE

A Dissertation
Presented to
The Academic Faculty

by

Hina Shah

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
College of Computing

Georgia Institute of Technology
August 2014

AN ETHNOGRAPHICALLY-INFORMED ANALYSIS OF THE INFLUENCE OF CULTURE ON GLOBAL SOFTWARE-TESTING PRACTICE

Approved by:

Dr. Nancy Nersessian,
Committee Chair and Advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. Mary Jean Harrold, Advisor
School of Computer Science
Georgia Institute of Technology

Dr. Gregory Abowd
School of Interactive Computing
Georgia Institute of Technology

Dr. Alex Orso
School of Computer Science
Georgia Institute of Technology

Dr. Colin Potts
School of Interactive Computing
Georgia Institute of Technology

Dr. Wendy Newstetter
External, Department of Biomedical Engineering
Georgia Institute of Technology

Date Approved: May 02, 2014

To my advisor, Dr. Mary Jean Harrold,

for believing in me more than I believed in myself;

to my caring parents, parents-in-law, and Alpa didi,

for supporting me through this journey;

and above all, to my loving husband, Samir,

for being my strength in making our dream come true.

ACKNOWLEDGEMENTS

My accomplishments throughout the Ph.D. journey, which included its fair share of highs and lows, could not have been possible without the support of many of my family members, friends, mentors, and loved ones. Acknowledging all those who made this Ph.D. journey successful is just one way to express my gratitude towards them for dispensing their support and blessing.

Its true that one's Ph.D. academic advisor is an academic parent. I have been fortunate to have two terrific advisors—Dr. Mary Jean Harrold and Dr. Nancy J. Nersessian—as my academic mothers and role models. They has been my source of inspiration, determination, and power. I am indebted to Professor Mary Jean for her unmatched dedication and relentless effort in supporting and training me to become who I am today. Although her absence has left a void in my life, she has left behind incredible memories and valuable lessons that I will always treasure. I would like to thank Professor Nancy for her excellent support, understanding, and inspiring leadership. Her expertise and exceptional guidance was a significant contributor in shaping such an interdisciplinary topic of research and successfully performing the ethnographic investigation. I will always be grateful to both my advisors for sharing their personal and professional experiences, which became a guiding light for me throughout my years of study.

I have been fortunate to have a distinguished set of professors on my defense committee: Dr. Gregory Abowd, Dr. Wendy Newstetter, Dr. Alex Orso, and Dr. Colin Potts. Many thanks to my committee members for their insightful guidance, much appreciated directions, and support. In particular, I am extremely grateful to Professor Wendy for providing me her time to have many insightful and intriguing research discussions and offering

me invaluable advice based on her expertise and experience performing ethnographic studies. Without her indispensable guidance, I would not have been able to fully appreciate the essence and deep-rooted power of this qualitative method. Also, I am grateful to Lisa Osbeck, Mike McCracken, Karen Head, and Bradd Shore for their feedback and advice, which helped me expand my knowledge and learning during the study period.

I am indebted to many members of the Aristotle Research group including but not limited to Saswat Anand, George Baah, Sarah Clark, Shauvik Choudhary, Jake Cobb, Sangmin Park, and Raul Santelics. Their moral, emotional, and intellectual support played a major role in making this Ph.D. journey a success. Moreover, my first family of friends at Georgia Tech included Nikhil Almedia, Krimy Amichandwala, Kedar Gaonkar, Nikunj Nemani, Rushabh Shah, Riddhish Shah, Varun Shah, Vinit Shah, and Rohan Vora; and to them I owe a huge debt of gratitude for all their support, care, and love. In addition, I want to acknowledge my "ethno-comardes"—Sherif Abdelmohsen, Altug Kasali, and Vrishali Subramaniam—who made my passion for such a research topic more fun, intriguing, and interesting. The lengthy discussions with them, ranging from philosophical ideas to practical challenges, in performing the ethnographic analysis, coding, and validation provided a terrific platform for learning and sharing. Moreover, many thanks to Bryan Wiltgen and Klara Benda for their invaluable feedback during the defense, which helped me clarify my presentation material leading to a better presentation. Finally, I want to thank my relatives, Shweta and Amish Kania, and Pranay and Sharmila Shah, and my loving little sisters Pragya and Shrestha for being an extended family to me and making me feel at home in Atlanta.

There are some members in my friends and extended family circle, whose support has been with me even before this Ph.D. journey began. I am greatly indebted to my friends Prashant (Amol) Shah and Dr. Abhijeet Vadera for it was their timely guidance that directed me to make some correct decisions in my academic life. Thanks to my friends from India—Samarth Patel, Ashok Lalwani, Mayur Nachnani, and Nidhi Pardasani—for believing in

me and supporting me during some tough times. In addition, my uncles Sunder Agrawal, Jyotindra Khatlawala, and aunty Chaya Khatlawala have always bestowed their blessings, which became my strength on several occasions. Moreover, I want to acknowledge my mentor Sunil Joglekar for strongly believing in me and bolstering my morale with his kind and encouraging words time and again.

The participant groups form the supreme determinants of qualitative research studies such as the one I performed in my Ph.D. research. I am deeply grateful to my participant organizations and the participant members for their cooperation, time, and trust in me. It was because of their support and cooperation that I was able to perform an uncompromised research investigation.

My Ph.D. achievement has been a common dream that many members of my family shared. I am truly short of words here to express the gratitude towards my family for being an undeterred support system for me. I want to thank my parents, Bhupendra and Tarunika Shah, and darling sister (didi), Dr. Alpa Shah, for having faith in me and having the confidence in my capabilities. I do realize that this success would not have been possible without my parents' countless sacrifices, sleepless nights, and unmatched effort to provide me with a positive and encouraging environment to flourish. Without my sister's tough love, motherly care, and sibling support my successes and achievements would have been impossible. They were the ones who seeded this dream and made it happen. Words cannot describe the deep gratitude I extend to my dear brother-in-law, Dr. Akash Shah, for understanding my struggles, providing me with constant encouragement, and always being my hidden strength. Many thanks to my parents-in-law, Jayu and Shivling Jamkhande, for being so patient and understanding with me and providing me with every kind of support I needed to pursue this dream. Thanks to my brother- and sister-in-laws (Dr. Amol Jamkhande, Dr. Sharmila Jamkhande, Nikhil Jamkhande, and Minal Jamkhande) for being very supportive throughout the journey.

Finally, I dedicate this work to my dear husband, Samir Jamkhande, for his matchless

support, love, and encouragement throughout the period of my study. He sacrificed many family moments and always gave my studies higher priority over other expectations. He has always been there to celebrate my achievements, bear with all my complaints, brainstorm my research ideas, listen to my thoughts and frustrations, and above all sail through the difficult times with me. I have and always will admire him for being a fierce protector of my emotions, executor of my dreams, and a loving partner in every sense.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Global software testing (GST) is a large industry that is growing over 20% annually. A survey of 200 industrial practitioners found that, among software-development related activities, software testing is the second largest outsourced activity (reported by 81%). However, despite the perceived benefits of global software development, there have been concerns about the quality of offshore work. A 2005 study of 5000 executives by Ventoro reported that 75% of the 68% of executives who were unwilling to offshore their software development mentioned quality issues as the primary reason for their reluctance. Studies have indicated that miscommunication and cultural incompatibilities are among the top reasons for offshoring failures [154]. The global software-testing practice demands that test engineers at the vendor organization adopt their client's (or potential users') perspective to perform testing tasks effectively. However, vendor test engineers and clients (users) have different cultures and thus, can have different understandings, perspectives, and expectations. These differences can influence the way testing is planned and performed, and thus impact factors such as the quality of testing (and product) and productivity.

Because global software-testing practice is both growing and it is thought that the quality of the work might be suffering due to cultural incompatibilities, it is necessary to investigate the meaning and role of culture in this context. However, many recent studies, which report on culture's influence on the global software practice, have used the now standard cultural-dimensions approach (e.g., Hofstede's) to investigate culture's influence in a manner that significantly limits the meaning of culture by restricting it to a list of pre-defined dimensions. Dimensional approaches to culture are used in global software engineering (GSE) studies to investigate the cultural influences and, consequently, they capture only

limited aspects of the impact that culture has on the individuals and surrounding environment of the practice. However, current research, within both the GSE and other communities, such as organizational science and management, lacks an understanding of the impact and significance of culture (particularly, national culture) on global teams [65]. Hence, it is necessary to study culture's influence on global software-testing practice by adopting a more open-ended approach so as to remedy the widely acknowledged limitations of existing cultural studies. The goal of this research is to explore in greater depth what and how cultural factors are influencing the GST practice by using the *culture-as-models* perspective to conduct a series of *ethnographically-informed* studies of vendor teams involved in GST practice. In this research, I have:

1. Argued for the usefulness of adopting a culture-as-models framework for analyzing the role of culture in GST.
2. Performed ethnographically-informed studies of vendor teams at three vendor organizations to gather insights into the role that culture plays in the global software-testing practice.
3. Analyzed the collected data to describe the global software testing practice and capture cultural models embodied in the practice.
4. Developed narratives showing how the cultural models interplayed with the global software-testing practice.
5. Presented the discussion and implications for practice and future research based on the findings and analysis.

By capturing cultural models embodied in the global software-testing practice, this research has (1) provided insights into the role of culture in the global practice, (2) contributed to a better understanding of the supports for and barriers to global software-testing practice, and (3) facilitated designing solutions to address the cross-cultural challenges in global settings (e.g., cross-cultural training programs).

CHAPTER I

INTRODUCTION

1.1 Motivation

Global Software Engineering (GSE), particularly offshored, outsourced software development, is a large industry [12] and it will continue to grow [28, 104]. A recent survey of 200 industrial practitioners found that, among software-development related activities, software testing is the second largest (after coding) outsourced activity: 81% of 200 participants reported that they outsource software testing [28]. Additionally, offshored, outsourced testing is currently a $25B industry and it is growing at an annual rate of over 20% [4].

The perceived benefits of this offshoring and outsourcing practice are reduction in costs, access to skilled resources, effective time-zone utilization, and improved quality of the work. However, a research survey in 2003 reported that 46% of the respondents to the survey considered the work of the offshore teams to be of "poor quality." Furthermore, an additional 14% of the respondents reported that their offshore teams' work was "unusable or a setback to progress" [28, 99]. Additionally, Ventoro[1] reported the results of a research study in 2005 with 5000 executives that shows that 75% of the 68% of executives who expressed an unwillingness to offshore their software development mentioned quality issues as the primary reason for their reluctance [147]. This study also reported that 36% of the respondents indicated that their offshoring strategy had failed and that "miscommunication and culture" was one of the top seven root causes for failure [147]. Moreover, Casey reports that "cultural differences have been identified as one of the five centripetal forces that negatively impact global software teams" [31]. Other studies have indicated that

[1] Ventoro was founded by seasoned executives from the offshoring, outsourcing world who recognized the challenges U.S. and European customers face when planning, implementing, and managing successful offshoring and outsourcing initiatives.

69% of all outsourcing projects fail, completely or partially. These studies report that the main reason for these failures is the lack of cultural compatibility between the vendor and the client and poor relationship management [3]. A recent Accenture[2] study reports that "miscommunication and a lack of cross-cultural understanding can hinder the effectiveness of global sourcing" [1]. The study also reported the top factors causing onshore-offshore problems were "different communications styles (76%), different approaches to completing tasks (53%), different attitudes toward conflict (44%) and different decision-making styles (44%)." Studies have shown that culture also impacts the functioning of specific GSE activities, such as offshored software testing [5, 31]. Wadia, senior global executive in Accenture's Network of Global Delivery Centers, commented in 2006 that "in our view, the physical obstacles to outsourcing such as telecoms and facilities have largely been resolved. However, the soft issues, particularly cross-cultural communication, will continue to present the main challenges to realizing global outsourcing's full potential" [1].

Some studies have specifically shown that culture impacts the smooth functioning of global software-testing activities [5, 31]. Other studies have discussed the influence of cultural factors, such as organizational culture, on the way in which software testing is perceived and practiced at the organizations [93, 113]. Thus, global software-testing practice is increasingly growing but the quality of the work may be suffering because of cultural incompatibilities. The impact of poor software testing on U.S. companies was documented in a NIST[3] study in 2002 that reports "estimates of the economic costs of faulty software in the U.S. range in the tens of billions of dollars per year" [136]. Because of the impact that poor testing has on U.S. companies that employ it, understanding how cultural incompatibilities may be affecting software-testing is crucial. Hence, it is important to study global software-testing practice and the influence of cultural factors on this practice [149].

Like any discipline, software engineering possesses its own culture. This culture has

[2]Accenture is one of the top global technology-services and outsourcing service provider companies.

[3]NIST refers to the National Institute of Standards and Technology, which is an agency of U.S. Department of Commerce and it has research laboratories that support federal research.

emerged over time and across nations, and is influenced by and embedded in other cultures (e.g., national and organizational). For instance, according to Fendler, the design and practice of software engineering as a discipline, in general, is highly influenced by western culture [50].[4] Additionally, with the increased practice of offshoring and outsourcing in the software industry, both national culture and site-specific organizational culture play significant roles in shaping the software-engineering practice at a site. Software-engineering literature recognizes and discusses the existence of these cultures and their differences. Many recent GSE-related studies have reported that cultural factors greatly impact the global software practice (e.g., [21, 22, 31, 46, 83, 152]).

However, most of these studies have used the approach of applying frameworks such as Hofstede's five cultural dimensions (i.e., power distance, uncertainty avoidance, individualism versus collectivism, masculinity versus feminity, long-term versus short-term orientation) and Trompenaars and Hampdeen-Turner's seven layers of culture (i.e., universalism/ particularism, individualism/communitarianism, specific/diffuse, affective/neutral, achievement/ascription, sequential/synchronic, internal/external control). Although these dimensions help researchers show that culture has an impact on the global software engineering practice, the notion of dimensions sends the message that culture's impact is restricted to only these dimensions, which is not the case in reality. The result is that the meaning of culture is significantly limited [124, 71, 95]. Moreover, these dimensions (specified in that research) do not fit well with how practice in specific settings is informed by and imbued by culture, and so make it difficult to explain the implications of cultural impact on global software engineering practice [21, 36, 124].

Given that culture plays an important role in global software (offshored) practice, it is necessary to conduct appropriate research studies (notably ethnographic studies) that explore, in more depth, the meaning of culture in this context and that will help to understand

[4]Fendler observed the difficulty of fitting the recommended software-engineering practice into African culture in Namibia [50].

the members' perspectives on practitioners daily routines [21, 36, 75]. Such studies help us understand better how and why culture affects the GSE practice and identify the cultural model of this practice [71, 124]. Such studies will also provide a rich understanding of the strengths and weaknesses of this practice so that culturally-aware recommendations can be made to improve it. Given that the U.S. is one of the leading practitioners of outsourcing, such improvement can greatly benefit the U.S. economy by reducing the cost and improving the quality of the software products that it develops.

1.2 Literature Review — Related Work

Researchers have performed qualitative studies in software engineering, and, in particular, in the global software-engineering domain. Additionally, researchers have studied culture in the global software-engineering domain. However, this previous research has limitations that motivate the need for the research that I have performed. This section presents an overview of this previous work. Because my research adopts an ethnographic approach, I start the discussion (Section 1.2.1) with some background on ethnography. Then, I discuss qualitative (ethnographic) research conducted in the domain of software engineering (Section 1.2.2). Next, I present an overview of the research conducted in the global software-engineering domain (Section 1.2.3). Lastly, I present a discussion of the culture-related research in the global software-engineering domain (Section 1.2.4).

1.2.1 Ethnography

Ethnography, a well-established qualitative method in anthropology, is centered on understanding in-situ practice from the perspective of people involved in the practice. Ethnography's emphasis is on describing a culture, and its goal is to capture (understand as well as describe through writing) the emic (insider's) view of the world from an etic (outsider's) perspective. "Ethno" means people and "graphy" means writing. Hence, the crucial elements of ethnography include not only the process of analyzing the setting being studied

but also the presentation of a "thick description" [5] of the experience lived (by the ethnographer) while performing the study [51]. As stated by Malinowski, the goal of ethnography is "to grasp the native's point of view, his relation to life, to realize his vision of his world" [90]. While conducting an ethnographic study, researchers conduct field studies by spending considerable time observing and engaging with their participants in their natural settings to gain insights into their daily work lives and cultures. The researchers use data-collection tools, such as participatory observations [130] and interviews [131], to gain these insights. The researchers then interpret and analyze the collected data and present it in the form of a "thick description" [51].

Ethnography was originally developed and popularly used by anthropologists to study radically varied and remote cultures [51, 90, 131]. More recently, researchers have started using ethnography to understand contemporary and accessible cultures including the corporate world. This method has been popularly used to understand organizational practices and people in action. For instance, Suchman, through her seminal ethnographic study of the photocopy machine, shows how users of the system make sense of the technological artifacts they use to perform their tasks [134]. She also argues that the real work is not always visible, so it is important to conduct ethnographic studies to make these implicit practices explicit [133].

Ethnography has also been adopted by other disciplines, such as cognitive science, for understanding human cognition through the understanding of cultural practices and complex social systems. For instance, Hutchins conducted ethnographic studies to understand human cognition through the study of many complex socio-cognitive systems such as the ship navigation and the airplane cockpit [71, 72]. He provides a detailed account of the

[5] "Thick description" is a popular term phrased by anthropologist Clifford Geertz (his book title uses it: *Thick Description: Toward an Interpretive Theory of Culture*); it refers to the detailed account of field experiences of the researcher who attempts to make explicit the hidden patterns and social relationships of a cultural group being studied and the context in which these patterns and relationships exist.

computational and social dynamics involved in operating these complex systems, and describes the distributed cognitive system.

More recently, ethnography has gained popularity in many other disciplines, such as information systems, human-computer interaction, and computer-supported collaborative work. The method is increasingly being used to understand different aspects associated with technical competences, skills, and system design requirements [18, 92]. One popular system-design-related ethnographic study is the study done by Bentley and colleagues of air traffic control systems [18]. Through this study, the researchers demonstrate the usefulness of undertaking ethnographic studies to understand the way to design an interface for the controller to the flight's database. Moreover, Sommerville and colleagues argue that the involvement of sociologists (in the form of ethnographers) in the system design process greatly benefits the design decision process [127]. Additionally, Hughes and colleagues discuss the benefits of ethnography to unearthing the "invisible" work-related activities that are crucial for designing appropriate computer systems for the users [77]. Their study illustrates an example of how, using ethnography, the researchers were able to determine the mismatch between the system functionalities conceived by the designers and its context of use. The authors discuss the new role—user's champions—that the ethnographers may take during the early stages of the system-design process. They also discuss the advantages of using ethnography to make the tacit and fine-grained aspects of work visible so as to make right decisions for designing systems.

Recently, anthropologists and ethnographers practicing corporate ethnography have provided a body of knowledge of the challenges and experiences encountered by them [37, 79]." For instance, Cefkin, in her book on "Ethnography and the Corporate Encounter: Reflections on Research in and of Corporations." The book discuss corporate-ethnography related difficulties such as reduced study duration (e.g. only a few weeks as compared to months) and access constraints and limitations (e.g., premise access permitted only for a short period) [80]. In addition to the challenges faced by researchers while conducting

ethnography in corporate worlds, there are also discipline-specific challenges. For instance, Robinson and colleagues report on their experiences performing ethnographic studies in the specific software-development industries [111]. In addition to discussing the benefits of using ethnography in the software engineering, Rönkkö presents the challenges that the software engineers encounter while adopting ethnographic methods for better understanding the software-engineering practice. Section 1.2 discusses research related to ethnographic work in software engineering in detail.

1.2.2 Qualitative (Ethnographic) Studies in Software Engineering

There is a disunity between software-engineering practice and theory [25]. Although much is written about the theory of software engineering, there is relatively less known about the way software-engineering practice is carried out in the real world. Hence, there is an increasing need to study software engineering in practice. To obtain this understanding of the real-world practice, researchers have now begun to adopt qualitative methods [132], such as interviews and observations, to study the practice. Qualitative methods, such as ethnography, have been used in other disciplines, including human-computer interaction and cognitive science. Such qualitative methods are being used to understand the natural settings and people's actions in these settings for understanding human cognition and informing system designs. However, only recently have these methods received recognition in the software-engineering community to understand various aspects of this discipline. The remainder of this section discusses qualitative research performed in the software-engineering domain by categorizing it based on the various aspects on which the studies focus.

Software Engineering Phases. Qualitative and ethnographic studies are now being performed to understand the various phases of the software-development life cycle, such as requirements gathering, system design, and system development. The focus of most of these software-engineering studies is to understand the challenges involved in the execution

of the phases rather than gathering information for the software being built. For instance, Rönkkö conducted an ethnographic study (using an ethnomethodological method) to understand the requirements-related problems encountered by practitioners in a distributed software-development project setting [112].

Markus and Keil, in their seminal qualitative study, discuss important design-related learning for developing information systems. Through their study, they showed that the system design failed its purpose because it did not consider the users' motivations [92]. Curtis et. al., conducted one of the first qualitative studies in software engineering to understand the problems involved in designing large software systems [41]. Based on the study findings, they also developed a layered behavioral model to explain the impact of some commonly existing issues on productivity and quality. These studies and others have shown the importance of understanding social contexts and settings in which the software (being built) will be used.

Software implementation (i.e., coding or programming) is the most widely studied phase in the software-development process. A wide range of studies have been done to comprehend various aspects of software-development activities and to understand software developers. Sillito, Murphy, and Volder studied the questions that programmers ask during the software-evolution process [125]; Ko and colleagues did work on understanding the information needs of collocated software-development teams [81]; and Robillard, Choelho, and Murphy investigated source-code usage by developers [109]. In addition, there are conferences dedicated to understanding aspects of programming, such as program comprehension (e.g., International Conference on Program Comprehension). More recently, Prior focused her dissertation research on providing a detailed account of the software developers' work practice. She identified relationships among the software product code, its infrastructure, and developers' work practice [106].

With respect to software testing, Whitaker argues that there is ample information related to the theory of software testing but little is known about how testing is done in

practice: "software testing is arguably the least understood part of the development process" [151]. As compared to other phases of the software-development life cycle, software testing is by far the least understood practice. However, there have been a few studies to specifically understand the software-testing domain. Martin and colleagues conducted an ethnographic study to show that organizational culture and context influences the way real-world software-testing practice is conducted [93]. Taipale and Smolander conducted a similar study to understand the practice and suggested improvements based on the study findings [135]. Also, Rooksby and colleagues conducted an ethnographic study of small, agile, open source, and academic in-house software-development teams to demonstrate the cooperative nature of testing activities [113].

The findings from these studies seem to convey the message that software-engineering is a people-centric process because people play a diverse and crucial role in the various software development phases. In other words, although the technical aspects of software development are important, the people aspects of software development are equally important and deserve further attention for improving the software-engineering practice.

Software Engineering Processes. Qualitative studies have also been used to better understand software-engineering processes. Prior, Robertson, and Leaney, through their studies, showed that the actual practice of software engineering differs significantly from the conventional recommended processes [107]. Also, there has been an increased criticism of the traditional processes (e.g., waterfall model) as being significantly process-centric. This criticism led to the invention of process models such as agile software development, which have been considered to be more people-centric processes. With the increased adoption of these newer processes in industrial practice, many researchers have now started studying aspects of these process models. For instance, Sharp and Robinson conducted an ethnographic study of extreme programming to understand the culture and community of an agile method [123, 110]. Moe, Dingsøyr, and Dybå, through their ethnographic study of a Scrum

team, identified some barriers to introducing agile software-development in practice [97]. Nonetheless, despite the popularity of agile methods, there still exists a large industrial population that follows more traditional process models for software development.[6]

In conclusion, there appears to be a preference shift from process-centric to people-centric process. However, because process-centric traditional methods are still being practiced in the industry, investigating such practices can facilitate better understanding of the people problems.

Non-Technical Aspects of Software Engineering. As discussed above, because they form an integral part of the software-development process, more recently, researchers have begun to draw attention to the *people* involved in the process. In fact, over the past few years there has been a workshop dedicated to understanding the "cooperative and human aspects of software engineering" (CHASE) which is held at the premier annual software engineering conference—International Conference on Software Engineering (ICSE) . Seaman, one of the earliest advocates for qualitative research, has done extensive work with her students on understanding non-technical aspects, particularly people's behaviors (e.g., [142, 143]. Additionally, Sharp and colleagues conducted studies to understand motivation factors among software engineers [121, 122]. Begel and Simon studied novice software developers to help with faster transitioning from novice to experts [15, 16]. Ko, DeLine, and Venolia conducted studies to understand the information that software developers in collocated teams need [81]. Also, Venolia, DeLine, and LaToza studied software developers at Microsoft to understand their work habits and satisfaction levels with the tools they used [146]. Aranda and Venolia described the coordination activities around bug fixing by studying the "life cycle" of a bug from the time it is reported to the time it is fixed [11]. Curtis et. al., developed a layered behavioral model, based on qualitative analysis, to explain the impact of some commonly existing issues on productivity and quality [41].

[6]In all my field studies, I observed that the participant teams I studied did not practice agile software development.

Software Engineering Tools and Techniques. In addition to the non-technical aspects, research has also been done to understand whether and how technology supports the software-engineering practice. Researchers have studied various aspects related to software tools, techniques, and associated technologies to better understand the software-development practice. For instance, Breu and colleagues conducted studies to better understand collaboration and tool usage between the developers and the bug reporters [24]. There is seminal work by Grinter on understanding how software-related dependencies create and impact the social relationships within the teams [54]. She studied the role of a specific tool— a software configuration-management tool—to describe the technical and social implications of the software-related dependencies. More recently, de Souza conducted ethnographic studies, for his dissertation, to understand individual practices of software developers who managed software-related dependencies (i.e., impact management) using several tools [45]. Based on this understanding, he developed a tool to facilitate and minimize impact-management-related activities and then evaluated the tool by conducting further studies. Additionally, researchers at Microsoft Research (MSR) are conducting extensive empirically-based research that includes analyzing socio-technical congruence, investigating bug reporting, and analyzing software-engineering artifact data through data-driven software-engineering research (e.g., [19, 55, 20]).

Cognitive Science Research in Software Engineering. There has been some work in the software-engineering discipline that describes the cognitive aspects of this discipline. For instance, Walenstein's dissertation research describes a theoretical framework for understanding and providing cognitive support for software-engineering tools. His framework is based on Hutchin's principle of distributed cognition [71, 150]. In addition, Aranda and Easterbrook discuss the strengths and weaknesses of using distributed cognition as a theoretical foundation for software-engineering research [10].

Global Software Engineering. One important dimension of software-engineering research relates to how the software teams are located. When the teams are distributed across various locations (within or across countries), the software-engineering practice is commonly referred to as Global Software Engineering (GSE). As this area is closely related to my research, the next section present a detailed discussion of research studies conducted in this area.

1.2.3 Studies related to Global Software Engineering

In the last decade there has been an increasing trend towards a globalization practice wherein the labor is distributed across various locations, including locations in different countries or continents. Primarily because of the perceived monetary benefits, globalization has gained momentum not only in industries such as manufacturing, but also in the world of information technology and computing. In fact, in the computing world, globalization is practiced in many different ways depending on the necessities and situations. With the increase in the availability of low-cost telecommunication and internet facilities, it has become possible for teams to work from different locations and still be remotely connected to each other. Such distributed development settings have now become a common practice.

However, with the increase in competition, companies are frequently faced with challenge of competing with other companies in the presence of tight budgets as well as shortages of time and skilled resources. To cope with such situations, companies are increasingly adopting the practice of distributed development, not only by distributing work across different locations within a company in one nation, but also by finding opportunities to partner with other organizations or open subsidiaries in low-economy nations. This practice of globalization, wherein software-development work is distributed across companies and/or nations is popularly referred to as Global Software Development (GSD) or Global Software Engineering (GSE). ACM published a study report on “Globalization and Offshoring

of Software" in 2006 that primarily focused on giving an objective perspective on current and future trends in the globalization of the software industry [12]. We define *outsourcing* as the process of one company seeking a service from another company (in the same country) [12]. Additionally, *offshoring* refers to the process of one company getting its work done in another country, irrespective of whether the work is done by a subsidiary of the same company at another location or another company entirely [12]. The company outsourcing/offshoring the work is referred to as the *client company* and the company providing the service is referred to as the *vendor company*.

In the past, there has been a debate concerning the benefits of adopting Global Software Engineering (GSE) practice. The perceived benefits of globalization include low-cost economies, cheap labor, easy access to skilled resources, optimum resource utilization, and follow-the-sun productivity. However, unlike the manufacturing sector that can be considered more machine-centric, software development is more people-centric. Hence, despite the benefit, there are hidden implications to the software-globalization practice that pose unique challenges and complexities. Challenges that have caught attention include handling and managing cultural, temporal, and geographical separation, lack of sufficient infrastructure, business property privacy, and national security [149, 12]. There are many industry reports, research papers, and other resources that discuss the pros and cons of adopting GSE practices such as outsourcing and offshoring [2, 141, 82, 38, 70]. In 2006, ACM conducted a study to provide an objective perspective on "the current and future trends in the globalization of software industry" [12]. They published a report that distinctly lists the benefits that GSE has to provide (e.g., access to skilled resources, cheap labor, and economic benefits). However, the report also acknowledges that there are some challenges (e.g., cultural differences, threat to national security and business property, and lack of appropriate educational policies) that need to be addressed to improve the practice.

GSE challenges occur mainly because of the distance caused by three differences — temporal, spatial, and cultural. Recently, researchers have directed their effort towards

understanding the challenges that are being introduced or amplified by these differences mainly because (1) GSE is now becoming an important and increasingly adopted practice in the domain of software engineering and (2) relatively less is known about how GSE in being practiced in the real world [149]. In a recent systematic literature review study conducted by Šmite and colleagues in 2010, the authors report that "Global Software Engineering is still immature" [149]. One of the research questions in their study is: "What is the state-of-the-art in empirical studies of GSE?" Based on the study analysis, the researchers insist that more empirical studies should be conducted to understand the GSE practice for designing appropriate solutions to address its challenges.

In the remainder of this section, I provide an overview of the research done in the domain of GSE. Although, there is interesting research going on in areas such as open source distributed development, agile distributed development, and distributed development related educational learning. I do not cover these areas in the literature review because my research focus is on understanding the global (i.e., distributed) software-testing practice in traditional industrial settings. Hence, these areas are not related to my research goals.

Technical Aspects. There is research being conducted to address the technical aspects of GSE challenges (e.g., understanding technical difficulties [78], building tools [129], and designing research agendas [53]). However, leading practitioners (e.g., Wadia, who is the senior global executive of Accenture's Network of Global Delivery Centers) of outsourcing and offshoring are of the view that, in GSE, while the technical issues are more manageable, the non-technical issues are the ones that impose significantly more challenge for the smooth operation of the practice [1]. This is also evident in the research topics in GSE. Most of the topics focus on soft issues such as collaboration, communication, culture, and management.

Communication, Coordination, and Collaboration. Considerably more research has been done in understanding the communication and coordination related challenges encountered in GSE because it has an high impact on the practice. For instance, Herbsleb and colleagues report differences in characteristics of the same-site and distributed communication and suggest some methods for speeding global collaborations [63]. They report that the communication networks and frequency of communication across sites is significantly smaller than within same site and work that spans across distributed sites takes two and a half times more time than same site. In another study, Cataldo and colleagues studied coordination breakdown problems in global software-development setting [34]. They argue that the availability of communication tools and processes may not be sufficient to achieve the necessary coordination required for globally distributed teams. Another study by Herbsleb and colleagues discuss different collaboration models [62]. Additionally, there are other studies that attempt to understand collaborative patterns [116], reasons for communication breakdowns [43], and strategies for improving collaboration effectiveness [56].

Tools Support and Usage. Coordination issues have given rise to another domain of research that deals with investigating and building tool support for addressing collaboration and communication issues. There is research done to address various aspects: investigating the use of instant messaging in overcoming collaboration challenges [48, 101], investigating the role of tagging [138], examining the use of the dashboard and feeds in collaborative software development [139], and assessing the use of text-based communication for requirements elicitation [26]. Additionally, researchers have designed new tools to support various aspects of GSE. For instance, Cataldo and colleagues developed a tool called CAMEL that helped with collaborative distributed software-design activities [35]. Furthermore, researchers at IBM Research, India, are exploring the usefulness of adaptive tool frameworks for enabling coordination in distributed software-development environments [126].

Management and Team Dynamics. There are many management related issues that are peculiar to GSE settings. Hence, there is research being done to understand these management-related issues in GSE. In the issues related to *people* management, Hinds and Mortensen have done work on understanding the role of conflicts in distributed and collocated teams [66]. Levina and colleagues have done research on understanding the interplay of boundaries and status differences in establishing effective communication in distributed teams [87]. Zhang and Shani have explored the competitive relationships that arise between the onshore and offshore teams [155]. Babar, Verner, and Nguyen have investigated the role of trust among software outsourcing practitioners in managing onshore offshore relationships [9]. In issues related to *knowledge* management, Manteli and colleagues have investigated the impact of multi-site software governance on knowledge management [91]. Also, there is some work done in the area of *project* management to understand the difficulties associated with distributed project management and solutions (e.g., [33, 42]).

Team Configuration and Structure Issues. Researchers have also conducted research in understanding different aspects related to desirable team configurations for global software teams. Ramasubbu and colleagues conducted an extensive study investigating the relationship between (1) team configurational choices and (2) productivity, quality, and profit outcomes [108]. Based on the study analysis, they provided managerial insights for team-configuration selection. Also, Solingen and Valkema conducted a controlled experiment to understand the impact of the number of sites in follow-the-sun software-development method on the individual's speed and accuracy of work [144]. Additionally, there is research that investigates the prefered locations for follow-the-sun software development [148].

Software Development Phases. Requirement gathering and elicitation requires the most amount of communication. Because face-to-face and synchronous communication is difficult to achieve in global settings, there is much research being done to understand and

address issues related to distributed requirement engineering. For instance, Laurent and colleagues build a visual modeling notation for assisting globally distributed requirements-engineering project teams [86]. Damian and Zowghi discuss the challenges in distributed requirements engineering because of cultural diversity, inadequate communication, and time differences.

Considerable research has been done in the area of distributed software development. For instance, Begel and others in the Human Interaction in Programming group at Microsoft Research have conducted studies to understand distributed software development related issues and subsequently built tools (e.g., CodeBook) to support development activities [14]. Lings and colleagues developed a reference model for supporting distributed development activites [88].

As mentioned earlier in Section 1.2.2 software testing, in general, has not received the attention that it deserves [31]. Hence, it comes as no surprise that little is known about global software-testing practice in the real world [31]. There has been some work on addressing the technical challenges associated with global software testing (e.g., [53]). However, Rooksby, Rouncefield, and Sommerville argue that "the determinants of successful software testing have little to do with purely technical considerations" [113]. Furthermore, they and other researchers emphasize that software testing should be considered a socio-technical activity and not a solitary activity [113, 31]. Based on this argument as well as the fact that social-cultural aspects play a significant role in the GSE setting, it can be argued that, to improve global software testing, it is important to understand the human and social-cultural issues encountered by software test engineers. However, this problem domain has been explored sparsely. To the best of my knowledge, there is only one systematic study, done by Casey, that concentrates on understanding how software testing can be carried out effectively in a globally distributed software-testing setting [31]. Casey used case study approach to identify key factors that hinder and facilitate effective execution of globally distributed software-testing activities.

Cultural Issues. Cultural differences have posed one of the significant challenges to the smooth execution of GSE practice. Recently, practitioners as well as researchers have realized culture's profound impact on the GSE practice. Hence, research is now being directed towards understanding the challenges imposed by cultural differences and providing solutions to address these challenges. In the next section, I discuss the culture-based research done in GSE.

1.2.4 Cultural Based Research in Global Software Engineering

Culture is deeply rooted in every individual and their everyday practice because culture moulds the way people think and act [102, 124]. Hence, it is no surprise that the everyday practice of software development is greatly influenced by culture (e.g., [85]). With the growing internationalisation of jobs, software teams usually have inter-cultural groups of members, distributed teams across different cultural locations, or both. In fact, inter-cultural dynamics are said to play a role at various levels, including the national, organizational, regional, and occupational levels [67].

As Olson and Olson precisely describe, "culture surprises" us with the myriad ways in which it influences a particular practice. [102]. Culture has been shown to influence many different aspects of the GSE practice and researchers have investigated these aspects to understand cultural influence on the global software practice. Brochers, through his personal experience, discusses how the cultural dynamics of the teams made it difficult and ineffective for them to implement some software-engineering techniques and best practices recommended by the community [22]. Winkler, Dibbern, and Heinzi describe a case study that analyzed the relationship between offshore outsourcing successes and cultural differences. They found that the role of the management has significant implications for ensuring successful GSE practice [152].

Studies have been done to show that culture has an influence on software-engineering education. Fendler and Winschiers-Theophilus argue that software-engineering practice as

a discipline is highly influenced by western culture [50]. Through the discussion of the struggles they had while teaching software-engineering principles in Africa, they demonstrate the failure of adoption of the believed-to-be universally valid concepts and methods of software engineering in African culture and argue that the "western values and perspectives [culture] are embedded in [the software-engineering] practice and further propagated in education." As a result, researchers such as Casey are investing effort in designing educational modules that teach students about the role of culture in the GSE setting [32].

There is research done to understand culture's impact on various aspects of software-development process. For instance, Hazan and Dubinsky have created a tightness model to analyze the acceptance level of a particular software-development method in specific cultures. They exemplify this model by showing the difficulties of fitting the extreme programming software-development method into Israeli culture [61]. In another study, Hestres discusses the influence of American culture on software design through the example of a case study of Microsoft Outlook [64]. One of the points he argues is that the user-centered design "process is strongly reminiscent of western humanism, which emphasizes the importance of the individual[s] over institution" [64, p. 17]. Casado-Lumbreras and colleagues explored the influence of culture on mentoring relationships and found that culture influences both formal and informal mentoring [29]. Hahn and Bunyaratavej investigated the influence of cultural dimensions on offshoring location sites and found that countries with higher levels of Hofstede's individualism and power distance and lower levels of uncertainty avoidance are able to attract more offshoring projects [57].

Another culture-based research category is associated with culture's impact on information and knowledge. Boden and colleagues conducted studies to explore culture's influence on knowledge management in globally distributed teams [21]. One of the findings of their study is that the team members tend to use popular cultural information to rationalize failures (i.e., they are inclined to draw conclusions based on national stereotypes when breakdowns occur). While Boden and colleagues investigated knowledge management, Vatrapu

and Suthers examined culture's influence on collaborative information sharing and organization [145]. They discovered that some cultures (Americans) were more likely to share strategies related to information sharing and organization than other cultures (Chinese).

Much of the past research has concentrated on showing the challenges that multi-cultural dynamics impose on GSE practice. However, recently, researchers are investigating the benefits of having multi-cultural group settings. For instance, Despande and colleagues investigated the weaknesses and strengths of culture in global software settings and discussed strategies that can help to take advantage of cultural differences in GSE [46]. Moreover, Casey discusses examples from a Irish-Malaysian GSE team study wherein the Irish teams were able to reduce the offshore attrition rate by leveraging some of the cultural difference factors instead of exploiting them as was being done earlier [30].

1.2.5 Limitations of Existing Research

Sections 1.2.2, 1.2.3, and 1.2.4 discussed studies done in various research directions related to software engineering, GSE, and culture. Although research is being conducted in various directions, research in understanding one aspect of GSE—culture's influence on global software testing—is missing.

On the one hand, it is important to study the *global software-testing* context for several reasons. First, software testing in general, and global software-testing practice in particular, are relatively less studied areas of research [113, 31]. Hence, there is a need to conduct empirical studies to better understand this practice. Second, there seems to be a common assumption that studies performed to understand software-development activities will suffice for understanding testing activities. However, the counter argument is that the basic foundational concepts of development and testing echo the fundamental differences embedded in these activities. Hence, it is worth investigating software-testing practice separately to better understand it. Third, it is assumed that outsourcing or/and offshoring software-testing related activities are smooth and easy [31]. However, concerns regarding the quality

of work offshore poses an urgent need to investigate, in detail, the manner in which global software-testing is realized.

On the other hand, it is important to study *culture in the global software-testing* context for specific various reasons. First, using a case study approach, Casey focused on the research question of identifying the "key factors, instrastructure[s], methods, processes, and procedures, which facilitate and hinder effective globally distributed software virtual team testing." Through his study, Casey showed that culture is one of the crucial factors that influences global software-testing practice. Hence, it is important to explore this thread further to better understand culture's influence on the testing practice.

Second, to the best of my knowledge, there have been no longitudinal ethnographic studies to understand culture's influence on global software-testing practice. Ethnography is a widely-accepted method in other disciplines, such as anthropology, for understanding culture-based factors. Hence, there is great potential to capture and expose hidden facets of cultural influences through the use of ethnographic methods. In fact, Boden and colleagues emphasize the need for conducting systematic longitudinal studies for better understanding culture's impact on GSE practice [21].

Third, in the research discussed on culture-based studies in GSE (Section 1.2.4), the majority of these studies use the dimensional approach to culture (proposed by researchers such as Hofstede [67] and Hall [58]). Recently, researchers have been expressing their opposition to the use of such dimensional approaches for performing cultural studies because these have several limitations (discussed in detail in Section 2.3). For instance, Irani and colleagues have argued that the perception of culture provided by dimensional approach is inadequate when compared with other, more open approaches [75]. Also, there have been several reports that address the limitations of dimensional approaches at various venues (e.g., work by Segal and others at the International Conference on Inter-cultural Collaboration [115]).

Hence, there is a need to conduct ethnographically-informed studies that use non-dimensional approaches to understand culture's influence on the relatively less-studied practice of global software testing.

1.3 Research Question

Given the limitations of the existing work, the goal of my research is to answer to the following research question:

> *What and how do cultural factors influence the way global (offshored, outsourced) software testing is practiced and what cultural models are embodied in this practice?*

Sub-research questions are

1. What cultural meanings are the vendor software-test engineers using to organize their behavior and interpret their experience with the client team members?

2. What tacit assumptions do software-test engineers make culturally when performing their daily activities?

3. How do the test engineers acquire cultural learning of their clients at the vendor organization? What interpretation/understanding do the vendor teams have of the client team's expectations? How are these expectations (the understanding of the testing tasks) conveyed, communicated, and negotiated between the client team and the vendor team?

4. What is the intended software-testing practice agreed upon between the client team and the vendor team? What is the current practice observed at the vendor end? Are the current practice different from the intended practice? If so, how?

5. What are the culturally embodied meaning of software testing for the vendor team members (i.e., general notion/thoughts they have internalized about testing)? How

are the global software-testing activities perceived by the vendor teams (i.e., the way the participants practice testing activities versus what seems to be expected)?

1.4 Overview of the Dissertation

This dissertation is organized into eight chapters. Chapter 2 discusses and compares two approaches to conceptualizing culture—the dimensional approach and the cultural models approach—and presents a reference framework that I designed based on the cultural-models approach, which is used to conduct the investigation presented in this dissertation. Chapter 3 introduces the method adopted to conduct the studies and presents the participant and study details.

Chapter 4 describes the customary activities that the global software testing practitioners perform on routine basis. Chapter 5 concentrates on the cultural practices that are embedded in the GST practice. This chapter primarily elaborates on two core discussions: (1) comparison of the vendors' perceptions of clients' cultural practices of testing and (2) narration of a cultural clash episode between a client and vendor team that occurred in one of my studies.

Chapter 6 and Chapter 7 present the findings resulting from the ethnographic analysis. Chapter 6 discusses the cultural models that I discovered to be embedded in the vendor-side of the GST practices that I studied. Furthermore, this chapter discusses how these cultural models interplayed, thus providing a potential explanation for the cultural clash episode discussed in the previous chapter (Chapter 5). In Chapter 7, I focus the discussion on one of the cultural models—Trust Cultural Model—and present the analysis of my investigation that explored the complex cultural system of trust and testing.

Finally, Chapter 8 presents the conclusion. This chapter discusses my reflections (Section 8.1), contributions (Section 8.2), and implications both for research and practice (Section 8.3.2 and 8.3.1).

CHAPTER II

CONCEPTUALIZING CULTURE: CULTURAL MODELS APPROACH

Culture is defined in many ways in the literature (e.g., Kroeber and Kluckhohn compiled a list of approximately 200 different definitions of culture [84]). As culture appears to be difficult to define, many researchers have tried to simplify the definition of culture by conceptualizing it in many different forms. In this dissertation, I purposefully do not define culture but rather concentrate the discussion on the approaches adopted to study culture. Consequently, in this chapter I present a framework for studying culture through cultural-models analysis based on Shore's idea of thinking of culture in terms of cultural models. The chapter discusses two different conceptions of culture—culture as dimensions (Section 2.1) and culture as models (Section 2.2), discusses the limitations of using a dimensional approach for studying culture (Section 2.3), proposes the cultural-models reference framework for conducting culture-based studies in GSE (Section 2.4), and illustrates the idea of cultural models through example in software engineering (Section 2.5).

2.1 Culture as Dimensions Approach

I refer to the culture-as-dimensions approach as the approach that views culture as a predetermined set of dimensions, which can be imposed on a setting to investigate culture's influence. In the GSE literature, the most common way in which culture is perceived is by reducing it to a succinct set of dimensions described by researchers such as Hofstede [67], Trompaneer-Turner [140], and Hall [59]. These are relatively more commonly-used conceptions of culture in the GSE community.

Hofstede's dimensions According to Hofstede, culture is defined along five dimensions. First, *power distance* relates to the extent to which members of the society accept and expect unequal distribution of power. Second, *uncertainty avoidance* relates to the extent to which culture governs the feeling of comfort or discomfort among members of the society in uncertain situations, and it relates to the degree to which the society tries to control the uncontrollable situations. Third, *individualism vs. collectivism* relates to the degree to which members of the society hold an individualistic approach or a collective approach where they coexist in unified groups, such as family. Fourth, *masculinity vs. femininity* relates to the emotional role distributions between the genders. Fifth, *long-term vs. short-term orientation* relates to the extent to which a culture governs its members to orient towards long-term prospects vs. short-term prospects.

Trompenaar's and Hampden-Turner's dimensions Trompenaar and Hampden-Turner describe culture using three layers [140]. The outermost layer, called *explicit culture*, is the explicit observable reality in the form of artifacts and products. The middle layer, called *norms and values*, relates to the mutual sense of right and wrong and of good and bad. The innermost layer, called the *assumption about existence*, relates to core levels of human existence and survival. Furthermore, Trompenaar and Hampden-Turner define seven fundamental dimensions of culture. First, *universalism vs. particularism* distinguishes cultures based on whether they hold an objective or subjective approach to factors such as correctness. Second, *individualism vs. communitarianism* relates to the degree to which the culture is individualistic vs. collectivist (i.e., Hofstede's third dimension). Third, *neutral vs. emotional* relates to the degree to which emotions are allowed to influence interactions. Fourth, *specific vs. diffuse* relates to the degree to which relationships have the personal vs. professional element. Fifth, *achievement vs. ascription* relates to the manner in which the society judges its members; cultures that exhibit the achievement dimension tend to judge their members based on recent accomplishments, whereas cultures that exhibit the

ascription dimension tend to attribute status to their members based on factors such as kinship, gender, and age. Sixth, *attitudes to time* refers to the way in which societies perceive time—sequential or synchronic. Seventh, *attitudes to the environment* relates to the way members of the society view the control to/by the environment.

Hall's dimensions For Hall, culture is communication, which consists of verbal expression (words), power and status expression (material things), and feeling expression (behavior) [59].Furthermore, he identifies key cultural dimensions that are classified based on time and communication patterns. Based on *time*, he classifies culture as monochronic cultures that perceive time linearly performing one activity at a time and polychronic cultures that perceive time more flexibly by allowing activities to be performed simultaneously. Based on *communication patterns*, he classifies cultures as high-context cultures, in which importance is given to the context rather than the content, and low-context cultures, in which importance is given to the content (information is exchanged through explicit and clear messages giving more data to reduce misunderstandings).

2.2 Culture as Models Perspective

In contrast to the culture as dimensions view, a more promising approach—culture-as-models— views culture as a process that creates and uses models, which need to be uncovered to understand culture's influence on a setting. Hutchins argues that "culture is a process, and 'things' that appear on list-like definitions of culture are residua of the process" [71]. Under this view, cultural dimensions are *products* of culture not culture itself. Hutchins adds that culture is an adaptive process that gathers partial solutions of frequently occurring problems. Shore also defines culture as a dynamic, on-going process that produces models (not amenable to list-like formulations), which are organized structures of thought and action that represent tacit shared knowledge that people acquire and use to interpret experiences and generate social behavior [124].

In *Cultural Models in Language and Thought*, the authors argue that culture is organized in the form of cultural models [69]. These models are "presupposed, taken-for-granted models of the world that are widely shared by the members of a society and that play an enormous role in their understanding of that world and their behavior in it [69]."

Building on this idea of cultural models, Shore further describes these models and their topology. According to Shore, these models may be individual (resides in the mind) or public (resides in the world), and they are constantly changing and are changed by the individuals and the environment around them. From this perspective, culture can be viewed as a heterogeneous (and large) collection of models wherein each model is a set of patterns that governs the "conventional behaviors" of an individual in a cultural community [124].

Models are classified into categories based on various aspects. The first classification is based on where the model resides—"in the mind" (internal) or "in the world" (external).

- Internal models, also known as *cognitive mental models*, reside in the minds of the individuals. These internal models are further divided into two categories. The first category is *personal mental models*, which are idiosyncratic models created based on individual's personal experiences and they are not shared in great detail with other members of the community (e.g., using my own mental map to remember directions to a place). The second category is *conventional mental models*, which are models that are internalized and transformed based on an individual's experience with the external social world and that form a part of the shared cognitive resources of the community (e.g., conventional gesture of Japanese head bow).
- External models, also known as *instituted models*, reside in the social world. Instituted models are the externalization of experiences in the social world. These models are recognized as being instituted models when they are "objectified as publicly available forms" (e.g., house structures and public spaces in the U.S. versus Europe) [124, p. 51]. This process of objectification of models as publicly available forms is known as *institutionalization*.

The instituted models are *internalized* to construct and transform the conventional mental models. The conventional mental models and the instituted models together form the cultural models, with the former residing in the mind and the latter residing in the social world.

The second classification is based on whether the models represent specific purpose models or more abstract schemas. Cultural models may be identified as *foundational schemas*, if they depict more general, abstract, and global representation than specific and concrete instantiations (i.e., models). However, this distinction is relative rather than absolute, and is useful in cases in which a set of cultural models share some common general characteristics.

The third classification is based on the perspective that the cultural models imply. Shore discusses two perspectives. The model created based on the actor's perspective (the *actor's model*) organizes the actor's experiences of the action in which she is engaged. The model created based on the observer's perspective (the *observer's model*) organizes the observer's experiences from a more neutral, outsider perspective.

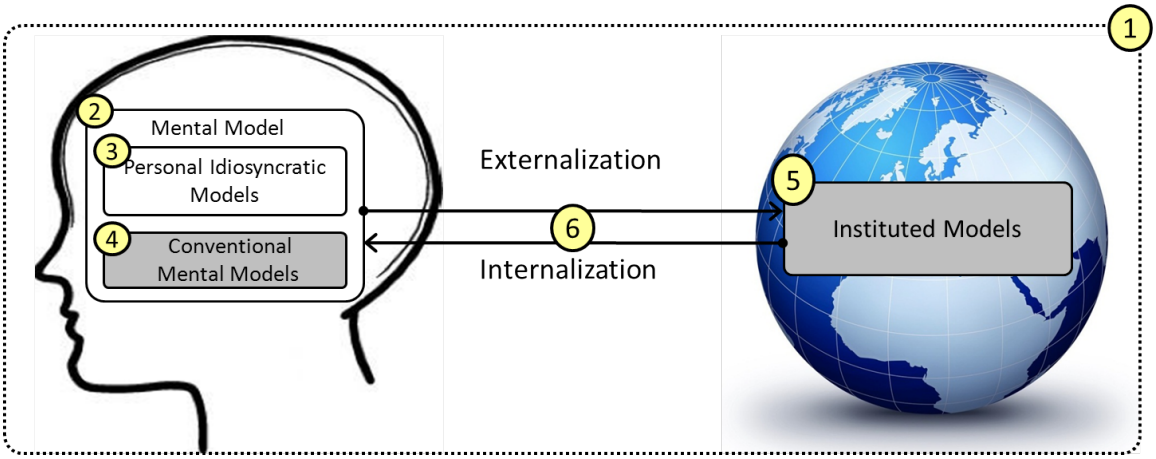


Figure 1: Culture as a Set of Models with Respect to an Environment.

Figure 1 shows my conceptualization of the cultural models discussion based on the classification of where the models reside. The models in the shaded boxes represent the cultural models. The numbers in the figure represent the different aspects:

1. Environment or cultural community boundaries
2. Cognitive mental models of the individual in that community (i.e., the internal model of the individual)
3. Personal idiosyncratic models created by the individual
4. Conventional mental models created in the mind through internalization of experiences in external social world
5. Instituted models that represent experiences externalized in the social world
6. Internalization and externalization activities between the two cultural models— conventional mental models and instituted social models

2.3 Limitations of Culture As Dimensions Perspective

The culture-as-dimensions approach to studying culture has several limitations [95] [49], especially when applied to GSE. In this section, we discuss these limitations.

2.3.1 Dimensions Not Embodied in the Practice.

The culture-as-dimension approach is being used in the global software-development community by applying the predetermined dimensions to the settings (or the situations being studied) and then comparing the results across settings (or situations). Using this approach, the dimensions do not emerge from the practice, and are thus, not embodied in the practice. Instead, the dimensions are applied externally to the practice. The problem with applying a predetermined list of factors to a particular practice is that there is a significant risk that cultural facets or features particular to that setting and that practice will be missed. To capture these features requires the use of qualitative and interpretive methods for examining the culture "from within." As Boden and colleagues argue, it is important to address these missing factors:

> The predominance of GSE [Global Software Engineering] studies that apply survey-type instruments utilizing forms of Hofstede's dimensions of cultural variation is problematic for the field, as it tends to over shadow the value of other, more interpretive approaches to understanding the myriad ways in which

> communication and coordination [and other cultural aspects] across globally distributed software teams is affected by local, organizational and professional 'cultures' [21].

Thus, to better understand culture's impact on global software engineering, it is crucial to capture the cultural factors embodied in the practice, which the dimensional approach currently fails to achieve, through the use of more qualitative and interpretive methods.

2.3.2 Dimension Pose Risk of Stereotyping.

As discussed above, researchers (practitioners) in global software engineering typically apply Hofstede-like dimensions externally to the practice to draw culture-based conclusions about their experiences in interacting with different cultures. However, there is strong argument against using such dimensional approaches to study culture because there is a risk of stereotyping [115]. Although, the use of such dimensions may help to gain a different perspective about the situations, it may cause the researchers (practitioners) to generalize their conclusions by stereotyping them. Such stereotyping is clearly described in the expatriate story discussed by Gertsen and Søderberg, in which one of the expatriates uses Hofstede's theoretical concept of power distance to justify why he feels at ease with the Americans [52] . Based on this story, the authors report

> What is interesting, though, is that the interviewee's use of one of Hofstede's concepts to explain his own affinity for the US leads him to talk about a colleague's difficulties in China, where he himself has never been, as a counterexample. The quote illustrates how a theoretical concept of a cultural dimension may be used discursively in different manners—to explain one's attitude based on presumed cultural similarity, but also to stereotype a culture with which one has not yet had any contact. [52, p. 254]

2.3.3 Meaning of Culture Significantly Limited.

Because culture is difficult to define, it is not surprising that there is inadequate understanding of culture in the first place, particularly in the global software engineering culture-based research community. In addition, dimensional frameworks' representation of culture tends

to misdirect many researchers into believing that culture revolves around these predetermined dimensions. Thus, researchers are inclined to explain any situation or behavior by fitting these dimensions to the situation or behavior, rather than attempting to determine other potential cultural facets that emerge from those specific data. Because of the manner in which these dimensions are being used to understand culture in the global software engineering domain, culture's meaning is being significantly limited. An understanding that culture encompasses facets beyond these dimensions appears to be missing. Imposing beforehand a set of dimensions along which to view culture in a particular situation or practice hinders the discovery of nuances specific to that culture that might be highly significant for understanding the problematic and supporting factors of that situation. This approach results in incomplete description of culture's influence on the global software practice.

2.3.4 Culture Viewed as Static.

With the use of the culture-as-dimension approach, culture is being viewed as a static and rigid entity that is represented by a set of characteristics embedded in these dimensions. Cognitive anthropologists, such as Shore and Hutchins, argue that culture is a process that is ongoing and is continuously influencing and being influenced by the environment and individuals in the environment [71, 124]. However, culture, in the form of numerical values of Hofstede's dimensions,[1] has not changed over time, which indicates that the culture-as-dimension approach misses the gradually evolving and transforming nature of culture. Hinds argue that "static entity approaches to culture [e.g., Hofstede's dimensional view] do not give a realistic picture of the means by which to address cultural differences in globally distributed teams" [39]. Thus, viewing culture in the form of such static dimensions misses the important aspects of culture as being dynamic and as gradually transforming in nature.

[1] Hofstede associates numerical values to his cultural dimensions for many countries (e.g., India's power distance index value is 77)

2.3.5 Culture Categorized Based on Various Boundaries.

Hofstede-like dimensional views of culture have created cultural categories (e.g., national and organizational) based on boundaries such as geographical and occupational. However, many researchers have argued against culture being restricted to these categories, especially the way it has been portrayed by the dimensional views of culture: as being purely national [74, 75]. It is difficult to disambiguate these cultural categories, especially in the global software engineering context, not only because of their highly overlapping and interconnected nature but also because of the increased practice of globalization and migration. For instance, global software practice has observed an increased migration of people around the world. Silicon Valley has many companies with the majority of employees being non-U.S. citizens. Consequently, people from different nationalities work together at one location and are distributed across different locations. Hence, instead of determining to which national, organizational, or occupational cultural category does a cultural facet belong, it will be more beneficial to identify the affording or clashing cultural models in the global software engineering practice.

2.4 Thinking and Acting through Models — Reference Framework

We speculate that one reason the cultural-dimensions approach (e.g., Hofstede's dimensions) has gained such popularity among disciplines, such as software engineering, is because it provides a framework for cultural analysis in the form of dimensions. Such dimensions can facilitate quick identification of cultural factors that are influential in a setting by providing pointers that guide the researcher in the direction of what to look for in the setting. However, as we discussed in Section 2.3, there are several limitations to using such a dimensional approach for studying culture. Hence, we propose a reference framework for studying culture through cultural-models analysis based on Shore's idea of thinking in terms of cultural models.

Figure 2 shows a conceptual framework of the cultural models and their connections in

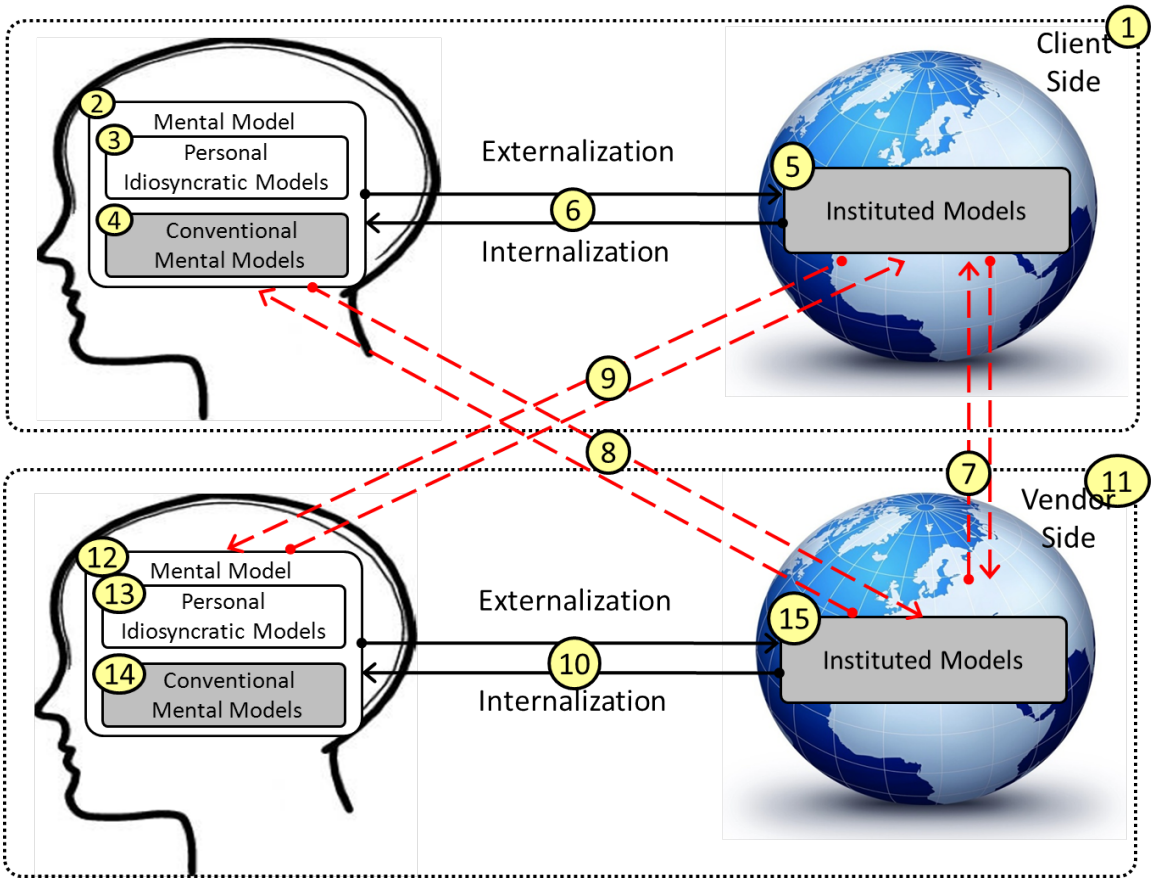


Figure 2: Cultural Models Reference framework for GSE Setting Studies.

a GSE setup. The solid (black) lines show the interactions that occur between the models, whereas the dashed (red) lines show the interpretations and comparisons that can occur. The models and connections in the figure are numbered and described as

1. Environment (community) boundaries of the client team
2. Cognitive mental models of the client team member in the client community (i.e., the individual's internal model)
3. Personal idiosyncratic models created in the mind of the client team member
4. Conventional mental models created in the mind of the client team member through internalization of experiences in external social world
5. Instituted models that represent experiences of the client team member, externalized in the external social world

6. Internalization and externalization activities between the two cultural models—conventional mental models and instituted social models—of the client team member
7. Comparison of the instituted models of the two communities—client and vendor
8. Interpretations of the instituted cultural models of the vendor teams, as internalized by the client team member
9. Interpretations of the instituted cultural models of the client teams, as internalized by the vendor team member
10. Internalization and externalization activities between the two cultural models—conventional mental models and instituted social models—of the vendor team member
11. Environment (community) boundaries of the vendor team.
12. Cognitive mental models of the vendor team member,(i.e., the internal model of that individual)
13. Personal idiosyncratic models created in the mind of the vendor team member.
14. Conventional mental models created in the mind of the vendor team member through internalization of experiences in external social world
15. Instituted mental models that represent experiences of the vendor teams, as externalized in the external social world

To understand culture's influence on global software engineering, a researcher may use the framework described above as a reference for his study. The researcher may conduct in-depth studies (e.g., ethnographic studies) of client and vendor teams to gather information for building the models shown in the figure.

For instance, a researcher can construct and compare the mental and instituted models at the client organization to study internalization and externalization processes within the client organization (e.g., connection 6) for understanding the influence that the organization has on the individual's activities and vice versa (i.e., studying internal organizational culture at a company). In addition, the researcher can construct and compare client team members' conventional mental models with vendor team members' instituted mental models and vice

versa (connections 8 and 9). Such comparisons may facilitate understanding of potential misunderstandings among client and vendor teams by providing insights into the expected and observed behaviorial patterns of the team members.

Thinking in terms of models using a reference framework like the one we illustrate can be beneficial in several ways. First, such a reference framework can help the researcher compare and contrast different models, which will help in understanding the similarities and differences among different cultures. For instance, a researcher can focus the study to create instituted models of software-test engineers job profiles at the client side and the vendor side. The client's instituted models may view testing jobs as highly-valued and prestigious (e.g., Japanese culture [153]), whereas the vendor's instituted models may view testing jobs as secondary, less-privileged [117]. Comparing these instituted models will help researcher understand cultural differences between the two communities, and subsequently reason about the consequences of differences in behaviors. Such comparisons can benefit understanding of which and how cultural issues need to be addressed so as to foster the relationships between client and vendor teams for better quality and productivity of work.

In addition, such a reference framework can provide a road map for identifying aspects that need to be studied further before drawing any conclusions about culture, and thus, avoiding the mistake of blaming culture for other issues (Damian and Zowghi report that sometimes blame was attributed to cultural differences to explain issues that were actually managerial [44]). Based on the framework and the research goals, the researcher may choose to perform an exhaustive study of a global software engineering setting by studying all aspects (all 14 aspects in Figure 2) or specific aspects (i.e., focus on a subset of the aspects). For instance, a researcher might choose to study the mental models of the communities because, according to Myers, software testing "involves some important considerations of economic and human psychology" and the mental models heavily contribute towards understanding human cognitive psychology [100].

Moreover, such a framework can help in accommodating the cultural transformations due to acculturation, because the difference between the conventional models and the instituted models of the individuals being studied will provide insights into the acculturation-related transformations.

2.5 Illustrating Cultural Models In Software Engineering

In this section, I explain the idea of culture as models by illustrating examples from the software-engineering discipline. The examples are mainly extracted from the studies I conducted (in my dissertation research) by reanalyzing the data collected in the field studies that I conducted at three multinational vendor organizations in India (discussed in detail in Chapter 3. The examples emerged from my reflections on several research questions including: (1) How are global software-engineering activities perceived by the participants at each organization? (2) Are these perceptions in line with what is being observed in the environment? (3) What is the cultural knowledge that is taken for granted in these global software engineering settings that I studied?

2.5.1 Software Engineering: A Modularized Culture.

I now discuss an example of foundational schema, which is a more abstract template of a collection of special purpose cultural models (refer to Section 2.2 for details). In his book's chapter entitled "Interior Furnishing: Scenes from an American Foundational Scheme," Shore discusses this *modularity foundational schema* that, he claims, has dominated modern American life [124]. Modularity schema is "breaking complex wholes into elementary units that are understood to be recombinable into a variety of different patterns" [124]. Shore provides an extensive illustration of the presence of this powerful and pervasive foundational schema in many modern American institutions, such as modular furniture, shopping mall layout, corporate modules, and educational systems. For example, he demonstrates how the modern sofa bed is designed in such a modular fashion that in one instance

it can be a sofa making the space a living room and in another instance it can be transformed into a bed making the same space a bedroom. The sense of a fixed form is divided into modular units that can be rearranged to create new configurations.

I observed a similar pattern in the software-engineering literature. In *Software Engineering: A Practitioner's Approach*, Pressman (an American) discusses different software-process models that guide the software-development life cycle. The modular nature of these models is distinctly evident in the way they are modularized into smaller units (i.e., framework activities, software-engineering actions, and task sets) [105, p. 78]. Any process model, such as waterfall, is divided into modules of activities: requirements gathering, design, development, and testing. Each activity may be divided further. For example, similar to the software-development life cycle there is software-testing life cycle, which divides the testing phase of the software-testing life cycle into activities such as requirements elicitation, test planning, test-case development, test execution, test reporting, and project sign-off. For another example, the phases of testing are frequently categorized based on the lowest-level structure of the software that is being considered as a element for testing: unit testing focuses on statement-level testing, integration testing focuses on the module-level testing, and regression testing focuses on retesting the previously tested modules. Not only is this modularization culture evident in the software-engineering processes, it is also evident in other aspects of software engineering, such as software-development programming paradigms. For example, the object-oriented paradigm adopts the idea of thinking in terms of modular units of objects and classes that are contained in packages and that contain methods. These modular units can be rearranged and reused in other contexts.

This example illustrates how software-engineering culture absorbs the pervasive modularized cultural foundational schema. Given this knowledge, the adoption and rapid promotion of the idea of GSE (i.e., distributed software development in the form of outsourcing and offshoring) may not be a surprise mainly for two reasons. First, as stated earlier,

software-engineering culture is embedded in the modular cultural schema. Second, globalization (i.e., GSE) is thought to be an extension of globalization practice followed in the manufacturing discipline. This modular cultural schema underlies the great successes of manufacturing production lines in America. As Shore explains "The American industrial genius was for breaking down production processes into their primitive constituents and then maximally rationalizing production by conceiving of labor and time as module entities subject to manipulation" [124, p. 131]. Thus, this underlying cultural model seems to promote the concept of global software development in the form of outsourcing and offshoring, wherein these modular chunks of activities (e.g., development or testing) can be easily rearranged to be executed from another location because they are treated as individual units [31]. Interestingly, a report that the Study S3 participants shared with me distinctly stated that North America was the top buyer of independent testing services (66%), despite the fact that buying only testing service requires unbundling the testing from the development activities. Nonetheless, unbundling may not have been perceived as a challenge because of the internalization of the modularized culture model by the U.S.. In fact, this underlying cultural model may be the reason that the U.S. is the leader in using these outsourcing and offshoring services.

2.5.2 Owning rather than Modularizing.

In Section 2.5.1, I discussed the modularized culture of software engineering. I illustrated that the testing phases can be modularized based on different units of activities or the element (scope) of the software system being tested. However, based my study analysis, the participants seem to have internalized a slightly different conventional mental model than the well-known modularized model. Instead of testing phases being structured around the software's structure, for the vendor participants the phases were structured around three high-level phases: test preparation, test execution, and client management.

- The *test-preparation phase* begins when the teams receive the functional requirements of the system. During this phase, in addition to understanding the functional requirements, one of the main activities is assessing the feasibility of completing the testing tasks in the stipulated amount of time. Negotiations are possible during this phase and can be performed by discussing task feasibility, requesting more knowledge transfer, reallocating the tasks, or requesting additional resources. The test-preparation phase overlaps with the development phase. Here, the developers own the system in the sense that they have the latest copy of the software being developed. However, not all the teams had a well-defined test-preparation phase in their projects. In particular, one team's participants reported that they never had sufficient time allocated to the test-preparation phase because of the nature of the activities they were supporting.[2]

- The *test-execution phase* begins as soon as the development team delivers the software to the test engineers. Participants mentioned that as soon as this phase starts, they experience pressure and urgency to finish their testing activities as soon as possible. They stated that it is difficult to negotiate (e.g., request more time) or raise concerns or issues because time plays such a crucial role in the phase.

- The *client-management phase*, also referred to as a warranty period by the participants, begins when the testing teams, based on satisfactory test results, approve the software system by "signing off on it." The system is then delivered to the clients and a warranty grace period starts, during which any defects found by the clients will be supported for resolution by the vendor teams.

Thus, instead of these phases being divided into atomic modules (as might be popular in the software-engineering literature), the participants seem to have internalized these phases

[2]They were supporting testing of high-severity (levels 1 and 2) bugs. Such bugs must be fixed and tested quickly (e.g., within a few hours) according to the service-level agreement signed between the client and the vendor.

based on who currently "owns" the code. During the preparation phase, the developers own the code (i.e., the developers create the code); during the execution phase, the test engineers own the code; and during the client-management phase, the clients own the code once it is deployed and shipped to them. Moreover, this specific cultural model of ownership is more of the conventional mental model instead of personal idiosyncratic model because it seems to be commonly, but implicitly, shared among these testers. Such a model that describes the phases as perceived by the practitioners helps in better understanding the practice.

2.5.3 Software Testing as "Second Standard" Field.

I now illustrate an example of the cultural transformation process and the struggle associated with this transformation that one of my participants—a software-test engineer at one of the organizations I studied—underwent. This example is representative of many similar experiences shared by other participants involved in our study. In this example, I discuss a portion of the interview with the participant. During the interview, I asked the participant about her experience as a software test engineer. In her response, she exposes the cultural model of software testing present in the software-engineering community. In her own words

> If you go out in the market…there is a rumor that testing is not a very good field. Testing is considered as a second-standard [class] field…When you are a fresher [novice software engineer], when you enter [join] the industry, you do not want to make a career in testing, you want to start with development.

This conversation shows the taken-for-granted cultural model of software testing in the community is that it is not a desirable field. It is a commonly held belief, which many participants confirmed, that testing job profiles[3] are less-preferred profiles than other software-development profiles, and that test engineers are considered to be "second-class citizens" [117, 118]. The conversation clearly shows the reluctance that novice software engineers have to joining the industry as software test engineers mainly because of this strong cultural

[3] A *profile* of a job is the description of the responsibilities and expected activities performed for that job.

knowledge in the community that "testing is not a very good field." Furthermore, she expresses the inclination of these novice software engineers towards a software-development job profile and reluctance towards a software-testing job profile because of the preconceptions formed based on this tacit cultural knowledge: "Initially when I was into testing, I seriously did not like it because I was [had] more of a development mindset." However, the participant mentioned that her organization's work customs are such that the employees do not get to select the job profile that interests them. Instead, employees are randomly assigned job profiles and, consequently, she was forced to work as a software test engineer: "[the vendor organization] has a random pattern that anybody is assigned any technology or any expertise (i.e., job profile)." Interestingly, after the participant began working as a software test engineer, she started appreciating this job profile:

> So when I went into testing, I learned the practices—they are totally different. A developer's mindset is totally opposite to a tester's mindset. I have to actually break the code and they have to actually create the code. So initially, it was challenging to me because I was not very familiar with the practices... processes that are involved, but slowly I developed interest ... testing is actually challenging, because you are always pressurized [pressured] due to deadlines. A developer might take 20 days to develop a code but you have to test it within the time line and within the given deadline. You cannot break [miss] it. [The] developer can break [miss] the deadline, you cannot break [miss] the deadline. That is the major challenge as a tester.

This conversation exhibits the gradual transformation of the participant's cultural model of software-testing practice. Her internal model of this practice now is a departure from the instituted external cultural model of this practice. She expresses her inclination towards this new profile. Nonetheless, she admitted that, earlier, she had conceived the same instituted cultural model of testing as being "second standard:"

> I was of the same mindset, but I broke that box [internalized a new meaning of software-testing profile] and finally I developed interest in testing because actually it was not possible for you to switch the expertise...if you have been assigned testing, you actually make your career, you build up your profile in that particular expertise only. So slowly and steadily by getting to know about it, knowing various processes, I finally developed interest, and finally I like testing, I can say.

What is interesting here is that her discussion implies a need to justify why she started liking testing. Such a justification echoes her internal conflict because she has now started internalizing the software-testing-practice model differently from the presumed cultural model in the society. Eventually, she gives up on this conflict of clashing models ("broke that box and finally I developed interest in testing") and accepts her inclination towards this profile.

However, inclination towards testing may invite its own new cultural struggles particularly because of the strong cultural model of software testing as a "second standard" field in the community. These challenges are evident in the way a software engineer, who is a software tester for a prolong period of time, is perceived in some communities. For instance, one of the participant managers mentioned that, in the western countries (e.g., he had worked with a client in U.S.), people prefer to be testers for many years. He had met someone who was a software tester in a software organization for the last 30 years: "however, in India, if you are a tester for such a long time, then you are perceived to be not capable to do other roles [e.g., managing or leading a team] so you are a tester (S2T3.PM1)."

Consequently, there seems to be a struggle to change this popular cultural model of testing as being "second standard." I discuss a senior manager's effort to change this cultural perception by reorganizing some organizational strategies around testing activities [117]. During the informal conversation with the manager, he agreed that he also thought that test engineers were considered as "second-class citizens," and mentioned that he had been making a conscious effort to change the perception toward testing among his employees. He discussed some of the steps he had taken. For example, in team review meetings, he always questioned the development teams when many defects were found by the testing team (a culture that is currently followed in countries such as Japan [118]. For another example, he also closely tracked the delivery dates when the development team delivered the code to

the testing team to see whether the testing team was given sufficient time for testing properly (deadline slippage from development team's side was a common practice, leadings to deadline pressure for the testing team). The manager also revealed his plan to reform the way in which e-mail communication occurred among the clients, the development-team members, and the testing-team members to ensure that the testing-team members got the recognition they deserved. Typically, the clients communicate through e-mail directly with the development teams. Thus, with the completion of a successful deployment, the clients appreciate only the development-team members for their efforts whereas the testing-team members' efforts are not acknowledged. The manager was therefore planning to reform the communication practices to allow direct contact between the client and both the development and the testing teams, thus ensuring that both teams receive acknowledgement of their work.

Discussion with another manager shows the transformation process towards perception of the testing profile that he underwent:

> I feel positive about it [software testing]...if you look at any project, 40% of the cost of the project is given to testing. So it's a whole lot of responsibility, and good amount of importance is given to it nowadays. That was not the case 6-7 years before. Earlier, it was seen as a wasteful exercise. When I started my career ...it was seen as mandatory evil, like you need to have it. So since the last 5 1/2 years I have been with [the company], it seems like big companies have different notions. Since then I have seen testing in a better light, and it is considered a very essential part.

Despite the popular perception of the testing profile, the test engineers preferred not to view their job profile as being "second class." Instead, they viewed it as an important profile in which they considered themselves to be the representatives of the clients (or prospective users of the product). In this role, they are then responsible for ensuring not only that the product is developed according to the clients (or users) expectations, but also that the quality of the product is high. Moreover, they perceived that the testing profile gave them the power to make decisions about approving (or rejecting) a product by "signing it off" (Section 4.1.3). Additionally, they seem to have defined new roles for themselves, such as

that of an information provider (discussed further in Section 4.3.2). Therefore, contrary to the popularly held opinion that a testing profile is generally less preferred than other software-development profiles, the participants were enthusiastic about their job profiles and they had a positive attitude toward testing.

These examples illustrate the gradually evolving and transforming nature of culture (i.e., culture of software testing). The question remains, however, of why software testing is considered as "second standard"? Is it because software testing is the last phase in the software-development life cycle? Would it have made any difference if it were first in the phase? Is that the reason why design and development profiles are considered as higher-status job profiles in the community? Does the order of phases in the software development life cycle have such deep cultural implications in the software-engineering communities? Such questions need further investigation to better understand the implications of culture and to direct the changes in the communities in the right direction.

CHAPTER III

METHOD

In Section 1.2.1, I described the qualitative method of ethnography. In my research, I adopted this qualitative approach to conduct ethnographic investigation of culture's influence on the global software-testing practice at vendor organizations. I distinguish an ethnographic study from an ethnographically-informed studies based on the duration of the time spent in the field. Ethnographic studies, are longitudinal field studies typically conducted by anthropologist researchers who immerse themselves in the field for extended periods of time. However, when such extended periods of field studies are infeasible, because of time, budget, or logistics constraints, ethnographically-informed studies, also referred to as rapid ethnographies, are adopted as a viable alternative [96]. These studies retain the ethnographic essence of understanding the "insider perspective," while conducting the study in a reduced interval of time. In this chapter, I present a discussion of the study method and provide the study details.

3.1 Overview

This section presents a discussion of the rational behind selecting the ethnographic approach to conduct my field studies (Section 3.1.1), describes my background and its effect on the ethnographic investigation that I intended to conduct (Section 3.1.2), provides an overview of the studies structure (Section 3.1.3), and the participating organization's structure (Section 3.1.4).

3.1.1 Ethnographically-Informed Study

> The choice and adequacy of a method embodies a variety of assumptions regarding the nature of knowledge and the methods through which that knowledge can be obtained, as well as a set of root assumptions about the nature of

the phenomena to be investigated [132].

Because this research has been concentrating on acquiring an empirical understanding of the role of culture on global software-testing practice, a method that would facilitate such an analysis of the practice was required. The method for exploring culture's impact, in general, has taken the form of interviews, quantitative analysis, introspective speculations, and observations, all of which contribute to the empirical understanding of culture's impact [124]. However, studying the process of culture's impact on global software-testing (GST) practice can benefit from the ethnographic investigation that facilitates the understanding of a group's knowledge, values, beliefs and their interpretations of the world in which they exist [40]. While ethnographic studies have been done to investigate different aspects of software-engineering practice (refer to Section 1.2.2), to the best of my knowledge, similar studies investigating the GST practice are lacking.

In ethnographic investigation approach, the study is not theory-driven. Instead, the approach adopts a bottom-up inductive analysis with continuous comparison of the researcher's interpretations with the different sources of data, which the ethnographer collects through the process of inspection and triangulation. Grounded coding is used to analyze the collected data for building or testing theories and themes. One of the main differences between ethnography and other qualitative research methods is that, in ethnography, researchers emphasize understanding their participants' interpretations of the practice that they enact [114]. The researcher's focus is on understanding how members of a group think, feel, act, interpret experiences, and create social behaviors in their everyday practice [130].

Reaching this level of understanding requires a close involvement of the researcher with the participants for an extended period with the goal of understanding various aspects of their daily (routine) work practice, such as learning their (technical) jargon, observing their interactions, capturing their emotions, and studying the artifacts that the participants' use. These required features are well facilitated by the ethnographic method. In fact,

to acquire such a deep level of understanding, the ethnographer systematically records, organizes, and analyzes the socio-cultural cognitive system being studied to build a "thick description" [5] of the behavior or practice being observed [51]. This description is not only presents a narrative of human behavior but also provides the context and discourse within the society being studied in a way that renders that behavior meaningful to the observer (i.e., the researcher). Moreover, Eaton, in her anthropological dissertation research, investigated the appropriateness of ethnography to study global, multi-sited, contemporary settings such as the information-technology offshore outsourcing setting. She concluded that, despite the traditional use of ethnography to investigate a single site, ethnography can indeed be a suitable and recommended method for studying such multi-site contexts [49, pg. 118]. Hence, ethnography was the appropriate method for addressing the research question being investigated in this dissertation.

3.1.2 Indigenous Ethnographer

My background had several overlaps with the setting I wanted to investigate. I am an Indian, I have a software-engineering background, and I have worked at organizations similar to the ones that I was going to study. Given this background, I was an indigenous ethnographer set out to study a relatively familiar GST setting.

Malinowskian manner of ethnography recommends that an ethnographer who is a stranger to the community investigates a location to reproduce and represent his vision of the reality of that local culture. *Strangeness* here is argued to be a vital factor to capture the local meaning of the culture within the community. However, ethnography has grown to offer much more than just investigating a strange culture. With the increased connectivity of the globe, this concept of strangeness has been shrinking as familiarity has increased. Particularly, in corporate ethnographies that has gained considerable popularity in the recent past, the value of ethnographic approach has been recognized and appreciated. The contemporary ethnographers have started surpassing this constraint of strangeness to the community because familiarity has its own set of benefits to offer, when appropriate methods are used to address the strangeness issue. They have started discussing this concept of familiarity using metaphors such as "backyard ethnography," which has been a current topic of discussion in the renowned conference of Ethnographic Praxis in Industry Conference—EPIC [137].

For instance, Treitlier argues that familiarity contributes towards developing the cultural models that are essential for acquiring a deeper understanding of the cultural meanings residing in the community being studied [137]. She further argues that the focus of ethnographic work should then be towards introducing ethnographic thinking rather than emphasizing the use of a "pure ethnographic method." Based on these discussions currently existing in the ethnographic research community (i.e., EPIC), in the section below I situate my studies by enumerating the advantages and the potential pitfalls I encountered for being an indigenous ethnographer. Furthermore, I present a discussion of how I overcame the

shortcomings of being familiar to the settings I investigated.

3.1.2.1 Advantages

However, there are several benefits of being an indigenous ethnographer when performing corporate ethnography such as the one I conducted. First, when an outsider is doing ethnographic research in a corporate organization, typically the researcher has access to the organization for a restricted period of time. Hence, indigeneity saves precious time invested in understanding the environment, setting, and context. Second, being an Indian, I did not stand out as an outsider but was perceived easily as someone belonging to their community. This perception of belonging to their community helped me in quicker rapport building and gaining the trust of my participants both of which are crucial elements when conducting an ethnography. Third, from the software-engineering research perspective, there is a constant desire to identify potential pain points in the software-engineering practice that can be improved. Being an indigenous ethnographer with software engineering background helped me better understand the expectations of my participant groups as well as other stakeholders who would be interested in the study findings.

3.1.2.2 Potential Pitfalls

There are disadvantages of being too close to the setting being investigated, which may have imposed unknown blinders impacting the study and its findings thereoff. For instance, because of being considerably familiar with the setting, I might have not noticed and discarded remarkable details as being common sense in the community. One way I made such familiar ideas strange was by conducting the pilot studies that helped me unlearn some of this assumed knowledge, which was perceived to be commonsensical details. The previous two studies significantly helped me in making the familiar strange and instead adopt a reflexive lens to revisit those details. For another instance, my background as being a software engineering Ph.D. student at a renowned institute such as Georgia Institute of Technology

may have been intimidating to my participants. I was aware and conscious of such implications and hence I assumed the role of an intelligent incompetent by downplaying myself while mindfully ensuring to maintain the trust of my participants.

Thus, my background facilitated me to augment my experience to the rich data I collected on the field. The ethnographic training yielded the sensitivity and consciousness to address the risk of biasing my findings with my own experience.

3.1.3 Studies Structure

I had the opportunity to conduct three studies—Study S1, Study S2, and Study S3— at three different vendor organizations in India (refer to figure 4). Because I came from a non-anthropological background, it was important for me to get some training in ethnography. Based on Spradley's note that "the best way to learn to do ethnography is by doing it [131, p. 42]," I trained myself by doing several studies. Particularly, the first two studies, along with ethnography-focused seminar and discussion-group with other fledgling ethnographers, helped me obtained the required training to design and conduct my final study.

However, although the intent was to perform ethnographic investigation in all the three studies, in additional to my lack of experience, corporate-setting related constraints such as restricted extent and duration of access to the field site and senior-management's control over the study's execution imposed challenges in conducting the intended ethnographic investigations.

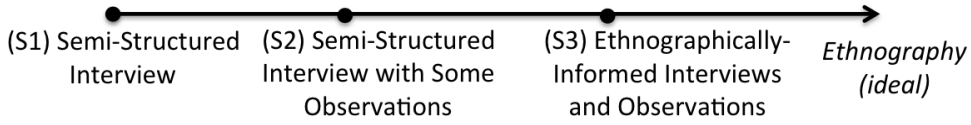


Figure 3: Gradually Evolved Spectrum of my Field Studies.

As a result, as depicted in the Figure 3, Study S1 was more semi-structured interview based study, Study S2 was also semi-structured interviews based studies but I had opportunities to conduct observations. However, Study S3 was relatively the closest to an ethnographic investigation, where I was able to conduct an ethnographically-informed study, after learning to better manage the constraints imposed by the corporate settings. In Section 3.5, I discuss the cumulative learning achieved through the three studies that I had the opportunity to conduct.

The studies were conducted at three different large Indian vendor organizations referred to as Vendor V1, Vendor V2, and Vendor V3, where vendor organization are organizations that provide software and IT consultancy services to various clients (e.g., U.S. organizations). At each vendor organization, I had the opportunity to interact with three teams (teams are referred to as T1, T2, T3). These vendor teams had different clients from different geographic locations as shown in the figure. In Study S1, there were clients from U.K., U.S., and local clients from India. In Study S2, two teams worked for the same U.K. clients whereas the third team worked for a U.S. client. In Study S3, the clients were from U.S., Mexico, and Japan. However, the U.S. and the Mexico clients were a part of the same parent company (indicated by the black bar in the figure) and hence they shared the similar organizational culture and policies. This parent organization is referred to as Parent (U.S.-Mexico) Organization. In this case, because the parent company was the same, the higher management (Delivery Management Level members in Figure 5) for the corresponding vendor teams were the same[1]. The vendor teams were working on client projects from variety of domains: Finance, Entertainment, Travel, Insurance, Retail, and Engineering. The nine teams and projects studied were all business critical projects but none of them were mission critical (i.e., belonging to aviation or health domain where safety policies are significantly strict).

[1]For logistics reasons, the vendor organization designed their delivery division's team structure based on the clients' team structure

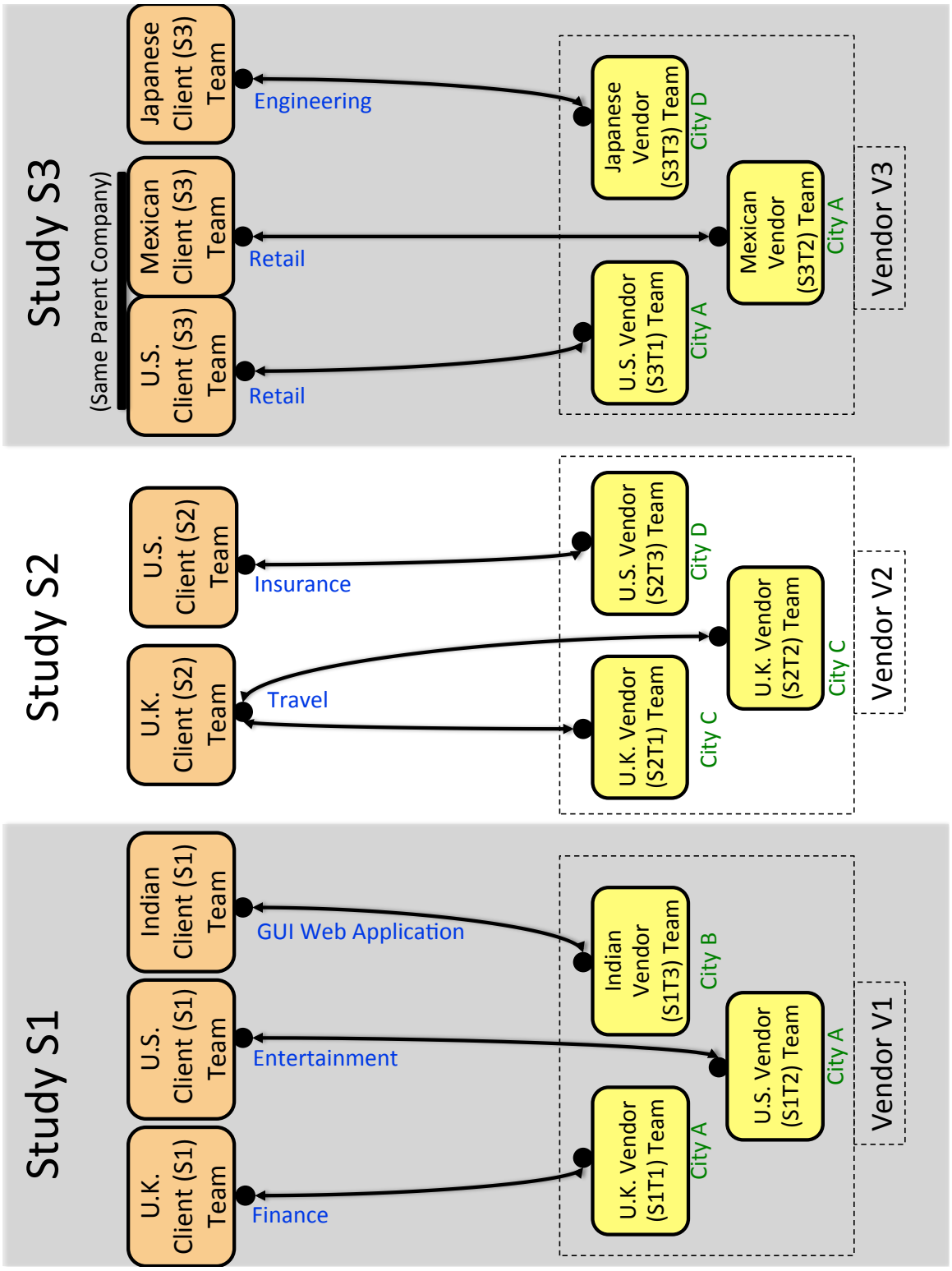


Figure 4: Study Overview.

Henceforth, I will use the following naming conventions to refer to the different study-related elements:

- Studies are referred to as S1, S2, S3, vendor organizations are referred to as V1, V2, and V3 and teams are referred to as T1, T2, T3.
- client team is referred to as U.K. Client (S2) Team, which means the U.K. client team from Study S1.
- vendor team is referred to as U.K. Vendor (S2T1) Team, which means the vendor team T1 from Study S2 who worked with the U.K. clients.
- parent organization for the U.S. and Mexico based clients in referred to as Parent (U.S.-Mexico) Organization

The team selection was done by senior delivery management (see Figure 5) at the respective organizations taking into consideration factors such as team's willingness to participate, their availability, and their respective project deadlines priorities. Nonetheless, I had provided three guidelines for team selection. First, the team size should be greater than five. A small team would not have helped gain the insights I was hoping to achieve. Second, the participating vendor teams must be involved in testing activities outsourced by some client team. The preference was given to the teams that were dedicated testing teams involved in providing testing services to the clients. However, a couple of occasions were exceptional—teams involved in both development and testing were also studied—because the team members had rich experiences that were foreseen to benefit this study. Third, teams should have deadlines for completing their testing activities. In total, I studied nine teams located across four different cities in India—City A, City B, City C, and City D (Figure 4).

All the three studies were reviewed and approved by the Institutional Review Board (IRB) at Georgia Tech. Thus, I ensured the privacy and confidentiality of the collected information and individuals' identities to the extent required by law and I also ensured that

the participants' work priorities, schedules, convenience, comfort levels, and privacy and confidentiality concerns are respected.

3.1.4 Organizations' Structure

The Figure 5 shows the organizational structure—divisions, groups within those divisions, and management hierarchy—in the Vendor V3. I interacted with members in the different divisions at Vendor V3 that are described as below. The structure is a typical structure of an Indian vendor organization and can be considered as a representative of the other two vendor organizations (i.e.e, Vendor V1 and Vendor V2) as well.

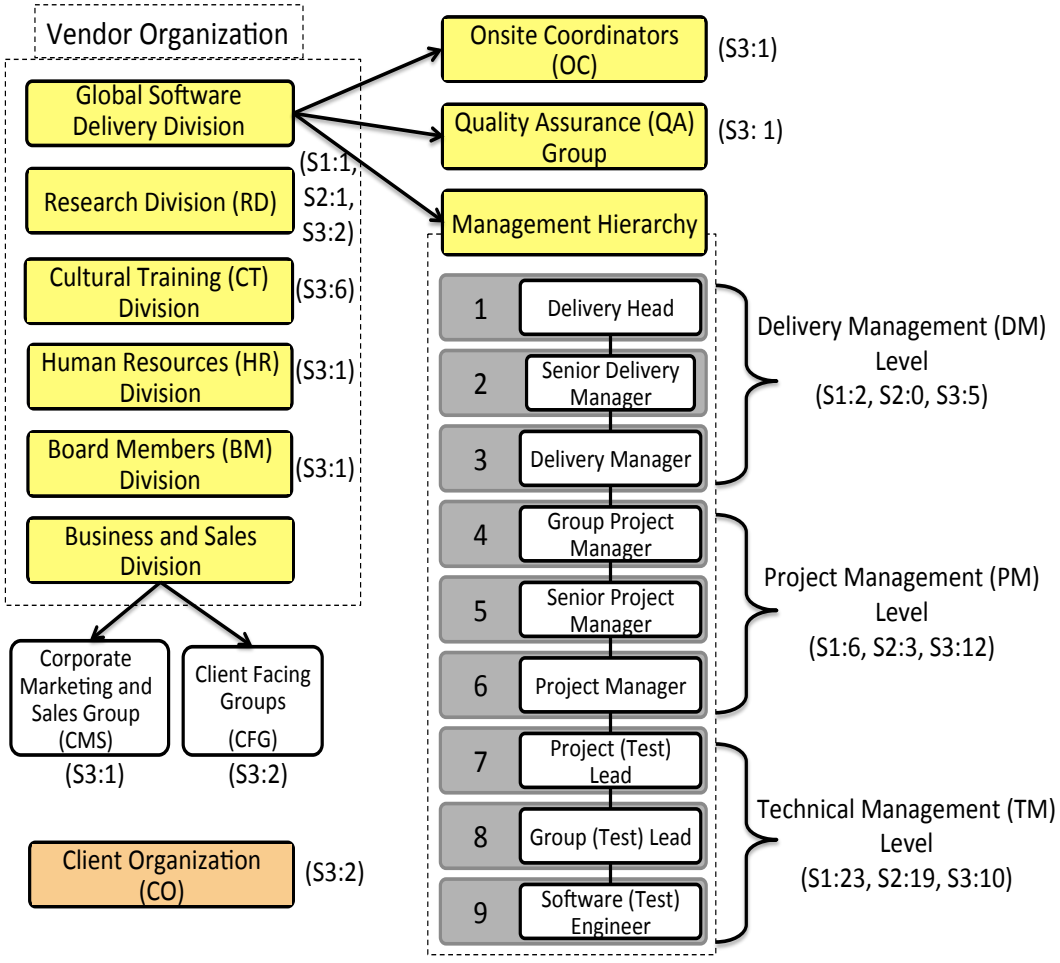


Figure 5: Organizational Structure's Overview.

Global Software Delivery Division. This division, which mainly provides GST service,

normally has a nine-level deep hierarchy. For simplicity sake, I categorized the hierarchy into three broad levels—Software Delivery Management, Project Management, and Technical—based on the commonality in the roles and responsibilities of the respective members within each level. For instance, the members at the Technical Management Level were responsible for managing the technical level details with respect to the project: the software test engineers were responsible for creating and executing the test cases (automation test engineers performed automation testing and manual test engineers performed manual testing), group test leads were responsible to decide scope of testing and oversee test engineers activities, and the project test leads were responsible to manage the technical issues and help the group leads in resolving issues.

Research Division. This division includes members responsible for performing software-engineering or related area research to advance the corresponding practice.

Cultural Training Division. The larger organizations in India typically have a cultural training division that provides cultural-awareness trainings.

Human Resources Division. My interest in this division was associated with the attrition related details for the teams that I studied.

Board Members Division. These members were the visionaries of the organizations and hence interactions with them shed light on the missions and goals of the organization.

Business and Sales Division. This division included marketing research team that was responsible to investigate new market trends with respect to GST practice, and the client facing group that was represented the clients at the vendor organization.

Onsite Coordinators. This group were the communication bridge between the client and the vendor teams and were responsible for collaborating between them.

Quality Assurance Group. My interest in this group was associated with the client feedback details for the teams that I studied.

Client Organization. I had the opportunity to interact with some members from the U.S. Client (S3) Team.

The initials in the parenthesis within each box in the Figure 5 are the identifiers used as a part of the naming convention for recognizing the different participants. For instance, S1T3.TM1 means a technical management level member from Team 3 of Study S1. Additionally, the text in the parenthesis outside the box indicate the number of members I interacted with in a particular study from that set. For instance, I interacted with 23, 19, and 11 team members from Study S1, Study S2, and Study S3 who were working at the Technical Level (e.g., Software (Test) Engineers, Group (Test) Leads, and Project (Test) Leads), which is indicated in the figure as "(S1:23, S2:19, S3:11)." For another instance, I had the opportunity to interact with only one onsite coordinator from Study S3.

3.2 Study S1: Semi-Structured Interviews Based Study

This first study was an exploratory study, where I concentrated on understanding global software-testing as practiced at the vendor organization and how human, socio-cultural facets influence the practice. I conducted a 35-day study, which consisted mainly of interviews, with some opportunities for observations, at Vendor V1. The next two sections discuss the study design and participants in detail.

3.2.1 Design Details

A group at Vendor V1 (henceforth referred to as the "study-design group") worked closely with me on designing and executing the study. Given that the study was to be conducted in an industrial setting, I prepared for it beforehand with the study-design group at Vendor V1. I held two telephone conference meetings with the study-design group from Vendor V1 to discuss the high-level goals of the study and the type of teams that were appropriate

for recruiting participants. Based on these discussions, I created a study proposal and circulated it among the study-design group members for their feedback. These discussions guided the study-design group in contacting appropriate teams to recruit them for the study.

Several days before starting the study, I met with the study-design group, as well as some other employees associated with the study-design group members, to discuss the study goals in detail. Based on the discussion, I listed ideas and focus points that might be worth observing. I then filtered the points to identify the key focal points for this study. The key focal points were agreed upon and finalized based on the interests of the researchers (i.e., my advisors and I) and the study-design group members.

The reminder of the section presents the details of the study that I designed in collaboration with the study-design group at Vendor V1. First, it describes the methods I used for collecting data for the study. Next, it presents the approach I adopted to create our interview guides. Then, it discusses our data-collection process. Finally, it presents the approach I adopted to analyze the collected data.

Because the focus of the study was to understand the human and social aspects of software engineering, my broad-level research question was

> *What are some of the 'people' problems that software-test engineers face while performing their work?*

Study Guide. For designing the study guide, I used the key focal points that I created during the initial preparation as my guidelines. I began by creating an unstructured study guide (refer to Appendix A). I used this unstructured study guide more to facilitate performing the study than to help directing the interviews. Because the guide was unstructured, it facilitated covering a large scope by letting me include additional interesting focal areas that I discovered during the data collection. As the study progressed, I created semi-structured interview guides that covered narrow but interesting topics that emerged during the study. Appendix A lists the questions that were included in the interview guide. Nevertheless, I was open to allowing the study to digress from my interview guide or key focal points when interesting topics caught our attention.

Study-Data Collection. I used the following procedures to collect the data:

1. I conducted an initial meeting with the participant teams (described above) followed by ongoing informal discussions (during lunch, breaks, or post-lunch walks) with individual members to get acquainted with them,
2. I conducted a series of interviews with participants and groups to gather relevant data,
3. I observed the individuals and teams, and attended the teams' group and client meetings whenever possible and permitted by the team lead. However, extended participant observation were infeasible in the environment at the three locations due to space constraints. Hence, the analysis largely included data collected from interviews.
4. I was unable to collect artifacts because of privacy regulations

The interviews and observations lasted between 30 and 45 minutes, and I conducted them with the consent of the participants. I conducted the interviews mostly in places other than the participant's working desk for several reasons, including space constraints, avoiding disturbance to other colleagues sitting close, or ensuring that the participants were in a comfortable location for sharing private information. Also, I did not audio/video record any of the sessions because the company did not allow such recording. Thus, the data I was able to collect in field were written notes.

Before starting the study with each team, I gave an introductory presentation and discussed the goals of the study, the plan for execution of the study, and the details of the privacy and confidentiality issues. Following this presentation, I arranged for an informal discussion that gave the participants an opportunity to ask questions or request clarifications. Given that the study-design group had contacted (for recruiting) the participant teams but I was the one actually conducting the study (without the presence of the study-design group members), this presentation proved useful for rapport building as it provided me with an opportunity to bond with the participants through informal discussions. I ensured all the

participants that I would respect their time, schedules, deadline pressures, and comfort levels for interviews.

Study-Data Analysis. I adopted a data-driven thematic approach for the analysis [60]. While I was collecting the data, I started identifying and listing interesting themes that were emerging across interviews. After the data collection, I read through the field notes multiple times to find evidence for the identified themes and to find new cross-cutting sets of common themes.

3.2.2 Participant Details

All participants in my study were involved in testing in some way (i.e., either they had the role of test engineers or they were in maintenance projects performing both development and testing).

Based on various factors, the study-design group identified three teams for the study—U.K. Vendor (S1T1) Team, U.S. Vendor (S1T2) Team, and Indian Vendor (S1T3) Team. The teams were located at three different offices in the company spread across two cities in India (refer to Figure 4 for detail).

Table 1 shows the details of the three teams. The columns, from left to right, list the team number, the type of project on which the team was working at the time of the study, the phase of the project at that time, the project's application domain, the locations where the clients resided, and the team size, respectively. The first project consisted of a five-member maintenance team working on an enhancement request received from their client, a U.S. based company in the entertainment domain. The second project consisted of a seven-member maintenance team working on test planning. The clients of this project were based in the U.K. and were from the finance domain. Because these projects were maintenance projects, the team members were involved in development as well as testing. The third project consisted of a 17-member quality-assurance team working on testing various Web applications for local customers. The structure of this team differed from the previous two

Table 1: Study S1 Participant Details

Team Name	Project Type	Current Phase	Domain	Client Location	Team Members
U.K. Vendor (S1T1) Team	Maintenance	Enhancement	Entertainment	US Based	5
U.S. Vendor (S1T2) Team	Maintenance	Test planning	Finance	UK Based	7
Indian Vendor (S1T3) Team	Quality Assurance	Test Execution	GUI Web applications	Local (India) Based	17

teams. This team was a dedicated testing team wherein all team members were performing only testing. With this third team, I also got an opportunity to interact with two members from their senior management (delivery management level). Thus, in all I interacted with 31 members in the study: two from the software delivery management level, six from the project management level, and the remaining 23 from the technical level. Out of the 31 participants, 15 participants had six months to two years of experience [2] and the remaining 16 participants had more than two years of experience.

3.3 Study S2: Semi-Structured Interviews and Observations Study

The second study (Study S2) was an (exploratory) ethnographically-informed study that concentrated on understanding how GST is practiced under deadline pressure situations and how the practitioners perceived the practice: their attitudes, values, challenges, and experiences. The challenges encountered in the first study, which I discuss in the summary section (Section 3.5), helped me design this study better I narrowed our study focus to the deadline-pressure situation because time is an important factor in testing practice [113]. Moreover, it was feasible to perform an in-depth study of only one situation given my two-month time limit at Vendor V2. Additionally, the topic of deadline-pressure had not been

[2]This criterion of two years distinguishes the experience level of the engineers as being novice (less than two years) or experienced (more than two years) and the criterion is based on (1) the participating vendor organizations' policies for identifying and recruiting employees having fewer than two years of experience as a junior (novice) and (2) my personal experience working at a company similar to the studied vendor organizations in the past.

investigated in the past and the participant organization was interested in exploring this area. This section describes Study S2's study design (Section 3.3.1) and the participant details (Section 3.3.2).

3.3.1 Design Details

The first study's experience had taught me some important lessons that helped me design the second study better. In this section, I present the details of this improved Study S2: the methods used for data collection, the approach for creating the study guide, the data-collection process, and the framework for analyzing the collected data.

Study Data Collection. Because the participant organization was a vendor organization that provides services to the client organizations, there were constraints on what and how I could collect data for this study. The constraints were imposed because of a confidentiality agreement between the vendor and the client. These constraints limited our ability to (1) record videos of the sessions, (2) take pictures within the work premises, (3) collect artifacts, and (4) access project-related documents (although the participants let us view the documents on their computers, they were reluctant to share any documents with us). Nonetheless, I had negotiated for permission to audio record the interviews before the study started (based on Study S1's experience).

I used four procedures to collect data: (1) I conducted 29 interviews[3] with the participants (resulting in approximately 29 hours of recorded data); (2) I gathered field data in informal discussions during lunch (i.e., I wrote the notes after the discussions), breaks, or post-lunch walks (approximately 11 hours of data); (3) we conducted occasional observation sessions (approximately three hours of data); and (4) I attended one participant team's group meeting (one hour). Holding extended sessions to observe participants performing their daily activities was difficult because of space, access, and time constraints. Thus, the

[3]I conducted followup interviews with some (but not all) participants.

study and the analysis was based on primarily ethnographic-interviews [131].

The interviews lasted between 45 and 60 minutes each, and I conducted them with the participants' permissions. I preferred to conduct the interviews mostly in places other than the participants' work desks but within their work settings (e.g., in conference rooms or the cafeteria) to ensure that the participants were comfortable while sharing private information and to avoid disturbing other colleagues. Additionally, I audio recorded some of the interviews but I was unable to record other interviews for several reasons: one participant was reluctant to be recorded, one participant permitted only a partial interview to be recorded, and I lost partial recordings of three participants because of recorder issues.

Before starting the study, I gave an introductory presentation to each team. The presentation included a discussion about (1) the goals of the study, (2) the plan for executing the study, and (3) the privacy and confidentiality protocols of the study. Immediately after this presentation, I arranged an informal discussion session for participants to give them an opportunity to ask questions or request clarifications. The senior managers had contacted (i.e., had recruited) the participant teams but I actually conducted the discussion session without the presence of the senior team members because based on my previous study's experience the participants did not speak up and and asked questions openly in the presence of their senior managers. This presentation were beneficial for rapport building: it provided an opportunity for an introduction to, and informal discussion with, the team. I assured all participants that we would respect their schedules, time, deadline pressures, and comfort levels during the study execution.

Study Guide. I was the sole researcher who performed all the data collection. For assisting the study execution process, I designed a study guide (shown in Appendix B). This study guide, which is similar to a semi-structured interview guide, consisted mainly of broad topics that were worth investigating (e.g., perceptions of pressure, communication,

and support). Although I designed general topics for discussion, the interviews were unstructured and the discussions were open-ended because I wanted to let the participants lead the discussions in directions that were most interesting to them. Consequently, the study had an ethnographic element because the primary focus was to understand the participants' perspectives on the activities they performed and what was important to them. Thus, I used the study guide as a support tool rather than a constraint for our study and to that front the study guide facilitated uncovering other interesting insights that emerged during the interactions with the participants.

Study-Data Analysis. I adopted the six-step thematic-analysis framework, proposed by Braun and Clarke [23], for analyzing this study's data. I adopted the generic thematic-analysis approach (instead of using the specific grounded-theory approach) because my goal was to identify and report themes, not to build a theory. The MaxQDA[4] application supported the analysis-related activities, such as becoming familiar with the data, organizing the codes, and performing the thematic analysis.

3.3.2 Participant Details

In Study S2, I had the opportunity to interact with 22 participants at Vendor V2 from the three teams—U.K. Vendor (S2T1) Team, U.K. Vendor (S2T2) Team, and U.S. Vendor (S2T3) Team—that agreed to participate in the study. In addition, one member from the research team at the organization also played the role of an informant while assisting me with the study logistics. Thus, in total I interacted with 23 participants in Vendor V2. The particular teams I studied were dedicated testing teams and their focus was on performing testing activities for client projects. All the participants had at least two years of experience [2] in software testing. Table 2 presents the details of the three teams, which are discussed below.

[4]http://www.maxqda.com/

Table 2: Study S2 Participant Details

Team	Testing Phase	Domain	Client Location	Number of Participants	Client Partnership Period
U.K. Vendor (S2T1) Team	Planning	Travel	U.K.	9	1.5 years
U.K. Vendor (S2T2) Team	On-demand Support	Travel	U.K.	3	1.5 years
U.S. Vendor (S2T3) Team	Execution	Insurance	U.S.	10	9 years

U.K. Vendor (S2T1) Team was involved in testing activities for the periodic software enhancements/releases of the client projects, and they were in the planning or "test-preparation" phase (as the participants referred to it) at the time of the study. The team worked on a project in the travel domain for a client based in the U.K. We interacted with all nine members of the team. This team was internally divided into two testing groups—manual and automation—based on the type of testing activities they performed. The vendor team had a 1.5 years of partnership with this client.

For this team, the vendor organization had an onshore (local) office in U.K. where the onshore team members were recruited and located. This onshore team, which mainly included local members from the U.K. and some members deployed from India, worked closely with the clients and played the role of an intermediary between the client and the offshore vendor teams. The industrial practitioners commonly refer to these intermediary teams as onsite coordinators. For this project, part of the development team was located offshore and the participant team frequently interacted with them. The U.K. Vendor (S2T1) Team (i.e., the offshore testing team) and their corresponding offshore development team reported their activities to the clients together.

U.K. Vendor (S2T2) Team was providing support for testing requests for high-severity defects, which are required to be fixed and tested in a short period of time. The role

of the team members was to provide continuous (24x7) testing support for system testing related activities. This team was working on a different sub-project under the same travel-domain project on which U.K. Vendor (S2T1) Team was working. The team worked with the same U.K. clients as U.K. Vendor (S2T1) Team but with different team members. Three out of seven members of the team participated in our study. The vendor team had a 1.5 years of partnership with this client.

The onshore-offshore team structure of this team was similar to U.K. Vendor (S2T1) Team. This team had the same offshore development teams as U.K. Vendor (S2T1) Team. For U.K. Vendor (S2T2) Team, the testing team and its corresponding offshore development team reported their activities to the client together.

U.S. Vendor (S2T3) Team was also involved in testing activities for the periodic software enhancements/releases of the client projects, and it was in the test-execution phase at the time of the study. The team worked on a project in the insurance domain for a U.S. based client. Ten of the 40 team members participated in the study. This team was also internally divided into two groups—manual and automation—based on the type of testing they performed. The vendor team had a 9 years of partnership with this client.

U.S. Vendor (S2T3) Team directly interacted with and reported to the client (without any intermediary onshore team). This team had an offshore development team. However, for U.S. Vendor (S2T3) Team, the testing team reported its activities to the client separately from its corresponding offshore development team.

3.4 Study S3: Ethnographically-Informed Study

This section presents the details of the design of the final Study S3. The goal of this element of research was to extend the knowledge about GST practice obtained from the previous studies and perform deeper investigation of the role of culture in the GST practice. After gaining sufficient training in conducting ethnographic studies, I designed and conducted

this final, comprehensive ethnographic study. The concentration of this study was on investigating questions such as what culturally-embodied meaning of software testing do the global teams (both client and vendor) adopt and what tacit assumptions do the teams possess while performing their daily activities? I adopted the ethnographically-informed qualitative approach to conduct the study. I conducted the four-month field study at Vendor V3 in India that provided various software-development-related services.

Section 3.4.1 discusses the study data collection details, Section 3.4.2 discusses the study analysis, Section 3.4.3 validation methods used for the validating the findings, and finally, Section 3.4.4 describes the participants of Study S3.

3.4.1 Data Collection Procedure

For the Study S3, I was again the sole researcher performing the data collection. The Vendor V3 allowed me to visit them and perform the study with the participating teams at the organization for approximately four months. The plan was to gather the required data by spending as much time as possible at the vendor organization. I was provided accommodation on the organization's on-campus housing for the duration of the study, which made the study execution convenient from several perspectives (e.g., observing meetings conducted at late hours with the client teams). I acquired prior permissions to get required access to the participating teams and other information (i.e., place to sit with the team, access to the organization's internal website, and permission to interact with the participants).

Conducting this study required regular visits to the participating teams. The duration and frequency of these visits depended on the nature of the participants' project, their availability, their deadline periods, the frequency of meetings, and the degree of access to the project's data. These factors were negotiated and determined with the selected project teams. During these visits, intensive data collection required a good rapport building with the participants. For building the rapport, I conducted introductory sessions with participant teams to get familiar with the teams, explained to them about the purpose of my research,

and provided them with an opportunity to clarify any doubts or questions. Such sessions have proved beneficial for building good rapport in the previous studies and were again quite useful during this last study. With permissions, I audio recorded the meetings and interviews to ensure the accuracy of collected data and to facilitate data analysis.

In the early phases of the study, I concentrated on understanding the participants' day-to-day activities and interactions, the organizational structure (within and between the various technical and non-technical teams), and the organization's goals and visions. As the study progressed, I directed the data collection effort to focus on aspects that were intriguing and that required further investigation and exploration. Through out the data collection process, the concentration was on (1) understanding the socio-cultural context of global software-testing work and (2) capturing common patterns of behaviors, interactions, thoughts, and activities.

For gathering the information, I used several different data collection procedures, which I describe below.

3.4.1.1 Ethnographic Interviews

I used ethnographic interviews as one of the primary data collection procedure, where ethnographic interviews are "a series of friendly conversations [in contrast to formal interrogations as done in formal interviews] into which the researcher slowly introduces new elements to assist informants to respond as informants [130, p.58]." The initial interviews were open-ended and they were conducted based on Spradley's guidelines [130]. These interviews concentrated on gaining an understanding of the context and environment around. Subsequently, semi-structured interviews were conducted with the participants. The interview guide was not pre-determined, but was designed and elaborated as the study progressed and it was continuously informed by the field observations and ongoing analysis. The guide also built on emerging data from other relevant sources. These semi-structured

interviews were mainly aimed to gain insights into the detailed specifics of the project's social setting, the participants' roles and behaviors in the project, their activities and responsibilities, the socio-cultural dynamics, and their interaction and relationships with others (including clients or other vendors).

In contrast to structured interviewing where questions are designed in advance and oriented to focus on specific points, open-ended and semi-structured interviewing procedure provided more flexibility to steer the discussions in directions that were interesting and needed further exploration. I audio recorded most of the interviews and observations.However, I was unable to video record because of lack of permission, and I lost one interview that I had recorded because of computer issues. Several follow-up interviews were conducted with the participants on regular basis to track progress on interesting issues, positions, and situations. The interviews were between 20 and 60 minutes long, and I conducted them with the participants' permission. I conducted the interviews in places that were convenient for the participants (e.g., their desks) except for a few interviews that I conducted in meeting rooms because I was discussing sensitive topics (e.g., client escalations and resignations).

3.4.1.2 Ethnographic Observations

Ethnographic observations were the secondary form of data collection (in other studies, it was difficult or impossible to conduct field observations for extended periods for several reasons including space constraints). This data-collection procedure included taking continuous field notes and memos from observations in the workplace. During the study I observed informal conversations within and across teams, interactions with the clients, clients' workshop when they visited India, and the teams' conversations with me (the researcher) over informal occasions such as tea breaks and lunch breaks. The documentation of these informal patterns of communication permitted in understanding of individual and collective positions of the participants within the team and across the organization. The

observations specifically concentrated on any interactions with the clients or vendor teams, any conversations within a team regarding the other team or interfaces between the two teams.

3.4.1.3 Meeting Sessions

Group meetings within or across teams provided another source of data, which were again audio recorded. This primarily depended on the schedule of the project and the frequency of the meetings. The study focused on observing three types of meetings: (1) internals meetings within the teams, (2) meetings between the clients and the vendors, and (3) organizational level meetings/trainings that were related to cultural skills training for employees. Figure 6, depicts the arrangement of a meeting conducted when the client visited the offshore site.

Figure 6: Meeting Session When Client Visited Offshore Site.

The first type of meeting provided insights into the dynamics of team members within the organization. This meeting helped in building cultural models of the teams from various perspectives (e.g., their perceptions about productivity, the counterpart teams, and quality). The second type of meeting helped gain insights into the relationships between the two teams, the interactions, and the comfort levels that the team members shared. This meeting helped in capturing aspects such as trust, compatibilities, and status dynamics and understanding the organization's perceptions about the client-vendor relationships. The third type of meeting provided opportunity to gain insights into the organization's approach towards socio-cultural issues and how the effort put in through various training sessions benefited the GST practice. For instance, in one of my studies, I learnt that organizations were providing training sessions for teaching ways to interact with the clients and learn about the client's culture. Based on further investigation, it was understood that the client teams paid particular attention to the communication skills and cultural nuances of the offshore employees and hence such training sessions were crucial for maintaining good relationships with the clients. I was fortunate to get connected with the training department within the organization. I was able to collaborate with them to understand the organization's approach to cultural awareness, training, challenges, and initiatives. They supported my study tremendously by allowing me to (1) observe various training sessions and (2) conduct follow up interviews with the trainers.

3.4.1.4 Focus Groups

In the past I tried focus groups with the entire team and observed that the subordinates did not speak up when their senior managers were in the same room. This led to dominance of the senior managers opinions' during focus group sessions. However, focus groups benefited when members sharing similar profile were participating together. These members typically had a meal/tea-break group also. So, I conducted focus groups with such groups when I wanted to understand a group's perspective on general topics such as the impact of

certain organizational policies on their work practice. These were typically opened ended discussions in informal settings and I took notes of the discussion post-meetings.

3.4.1.5 Artifacts and Project Documents Access

I collected various artifacts including exchanged emails, photographs of physical layouts (e.g., during team meetings), project documents, customer-feedback reports, and presentation slides of various meetings including client-visit meetings. The information obtained from these artifacts was primarily used for the validation method of triangulation (refer to Section 3.4.3.2) but it occasionally also served as a lead that helped direct further investigation on interesting threads that emerged during the study.

3.4.2 Analysis Method: Thematic Analysis Approach

This research adopted the inductive thematic-analysis approach described by Braun and Clarke for analyzing the collected data [23]. In contrast to the theory driven thematic analysis, the data-driven thematic analysis method follows the approach of identifying, analyzing, and reporting emergent patterns, phenomena, and themes from the collected data through a series of steps (as described in detail later in this section) [23]. In fact, Braun and Clarke recommend this method to be "particularly useful method when you are investigating an under-researched area, or you are working with participants whose views on the topic are not known," which is the case in this research [23, p. 83].

The goal of the analysis was to identify cultural modelsin the form of themes that "capture something important about the data in relation to the research question, and represents [represent] some level of patterned response or meaning within the data set" [23, p.82]. Thus, I wanted to adopt an method that facilitated me to use this framework for conducting my analysis. As against the popular grounded theory analysis [132] that aims at building a *theory*, the thematic-analysis method, which also adopts the grounded-coding approach, better suits the goals of this research because it facilitates the building of cultural models (using the framework) in the form of *themes*. Although finding themes is how a theory

is built, because the goal of this research was not to ultimately build a theory I found it reasonable to adopt the thematic analysis for such an analysis.

The basic idea of thematic analysis is to have the researcher immerse in the data by continuously examining and reading multiple sources of data such as field notes, interviews and memos. This is followed by discovering sets of concepts (i.e., codes) that emerge from these data sources and identifying relationships among them, which are further categorized and refined into themes that form the narrative.

The initial analysis process was generating codes—labels that are applied to raw data fragments that describe a particular concept. Coding occurred in parallel to the data-collection process and it informed and was informed by the data-collection process. Coding was done promptly for the transcribed data (refer to Appendix D for a sample transcript), it was build upon the previous codes, and it informed the subsequent coding process. The six-steps of thematic analysis framework, depicted in the Figure 7, are described below.

1. *Becoming familiar with the data.* The first step was to get familiar with the data. Here, the focus was to thoroughly read through all the transcripts and collected data. The picture shows a snapshot of the MaxQDA application, which was used for getting familiarized with the data and also to perform other steps of the data analysis.

2. *Generating initial codes.* The next step was to generate initial codes by collating interesting features of the data. The figure shows some example codes from the previous study: pressure perception, strategies adopted to handle pressure, and challenges encountered (refer to Appendix E for the coding guide).

3. *Searching for themes.* Once the initial codes were generated, next step was to arrange similar codes to form potential themes. One such theme that emerged in the previous study was execution phase pressure experiences (refer to Appendix E).

4. *Reviewing themes.* The next step was to check whether the themes work in relation to the extracted codes by generating many thematic maps, which are organized maps

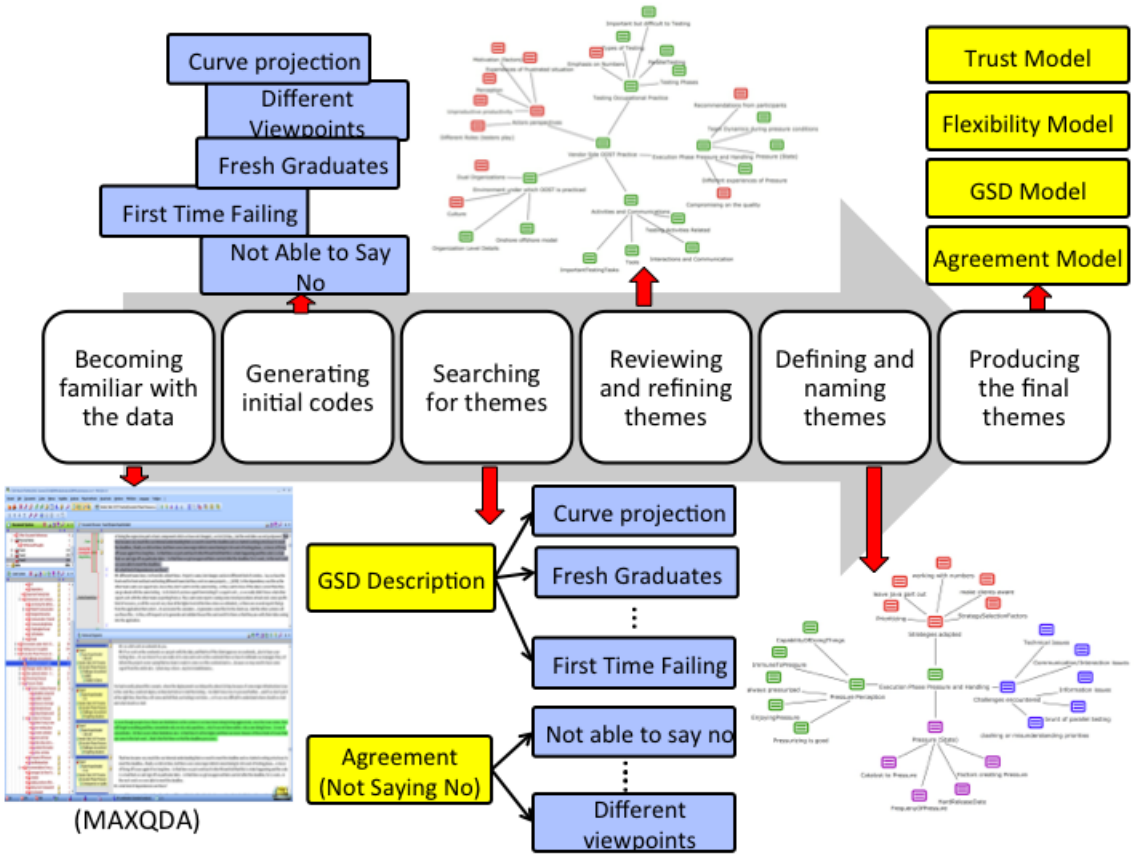


Figure 7: Thematic Analysis Steps.

of the codes and their relationships to each other. Using MaxQDA, thematic maps were generated like the one shown in the figure (these maps facilitated reorganizing the codes and rethinking the themes).

5. *Defining and naming themes.* The relationship among the themes were revisited and revised in an iterative refining process to precisely define and name the themes.

6. *Producing report.* Once the themes were finalized, the themes were related back to the research question and compelling and supporting data extracts were selected to produce a document (e.g., narrative) of the analysis findings.

3.4.3 Validation Method

There are several ways to evaluate the reliability of qualitative research like the one proposed here. To attain a high-level of reliability concerning this study's analysis (i.e., generating codes and building themes), this research implemented various approaches including inter-rater reliability, triangulation, and transferability.

3.4.3.1 Inter-rater reliability

Inter-rater reliability is the process of obtaining concurrence among coders concerning the coding method and the resulting codes. The goal of this process is to identify the degree of similarity of judgements, among independent coders, regarding the codes assigned to the collected data. When there is considerable agreement among the coders, there is high inter-rater reliability. For this research, high inter-rater reliability target was set to be 90%, which means that when the participating coders agree upon 90% of the codes, high inter-rater reliability will be said to have been achieved.

The inter-rater reliability was achieved as follows: I first read and coded the transcripts to identify the potential emerging patterns based on my interpretations of the data. Appendix D to a sample of the code transcript that emerged from the data collection process. Based on the codes, I developed the codes to form the initial themes. Another coder was recruited and the coder was presented with one of the transcripts that I had already coded and was requested to code the transcripts. At this point, my codes or themes were not shared with the other coder. Later, we shared our codes with each other, and discussed and negotiated over our codes and themes. Based on the outcome of the discussion, I revisited my codes and modified them accordingly. I then shared the updated coded transcript for the second round of negotiation. The initial level of agreementm when we started the process, was 66%. However, after two rounds of negotiation, we reached a concurrence level of (95%) with respect to the codes and themes at which point the discussion was ended, and the agreed codes and themes were used for evaluating the remaining codes. Appendix C

illustrates the instructions that I shared with the other coder who performed the inter-rater reliability.

3.4.3.2 Triangulation

The process of triangulation emphasizes the incorporation of evidence from various data-collection methods (methods triangulation), various data sources (sources triangulation), multiple analysts (analyst triangulation), or multiple perspectives or theories (theory/perspective triangulation) [103]. I validated my analysis using the triangulation of data sources method (other ways of triangulation were beyond our study's scope).

The triangulation process was mainly carried out towards the end of the analysis and coding process. It aimed at combining multiple data sources to view a situation or theme from various different perspectives. I compared the data from various sources (e.g., interviews, meetings, or documents) to check for the consistency of participants' thoughts on (1) various occasions (e.g., informal discussions, meetings, and one-on-one interviews) and (2) over a period of time (e.g., during various follow up discussions). This method helped to cross-check information shared by the participants and to eliminate their (as well as my) personal biases and interpretations. This process was conducted as follows: for every potential theme emerging from the data, effort were put in towards gathering evidences relating to that theme from various other data sources. If relevant data was not available from other sources, then effort were directed towards obtaining related information from external sources that provided more insights into the findings.

3.4.3.3 Member Validation

An important element of qualitative methods validation is confirming whether the interpretations of the researcher matches that of the informants. To validate these interpretations and my understandings, I conducted several follow-up meetings to get participants' feedback by presenting them my findings in various forms—by creating poster, diagrams, and

descriptive documents, and through face-to-face conversations. I performed these follow-up sessions with participants by emailing them, having online chats, or scheduling meetings (for lengthy discussions). Before the follow-up meetings, I carefully read through their previous discussions and documented my understandings and framed follow up questions (e.g., clarifications). Additionally, I asked participants to draw diagrams whenever applicable, or asked them to verify my diagrams to confirm our understanding.

Figure 8 illustrates an example of the drawings I shared with the participants for confirming my understanding during the study. The left topmost drawing is the poster chart that I maintained during the study. I drew the organizational structures, team structures, and other details as and when I collected data that clarified this understanding. The bottom left drawing was an externalization of my understanding of the roles and coordination among the roles. It showed who coordinated with whom based on what I understood in an interview with one of th participants. The last drawing on the right was used to confirm my understanding of the "kensa" testing group of the Japanese Client (S3) Team.

3.4.4 Participants Detail

During the study-execution period, I had the opportunity to directly interact with 44 participants but I had the opportunity to observe many more participants (e.g., during team meetings, talks, and training sessions). The study participants can be divided into two broad categories: project-specific and project-independent. The *project-specific* participants included those who belong to one of the three vendor teams in Study S3. Table 3 presents the details of the three teams, which are as discussed below.

U.S. Vendor (S3T1) Team. included members of the vendor organization who worked on the project for U.S. Client (S3) Team. They were in their test planning phase at the time of the study. The team was working on a project in the retail domain. I interacted with five members from the team. The partnership was three years old and during the three years the vendor team was responsible for performing testing (both manual

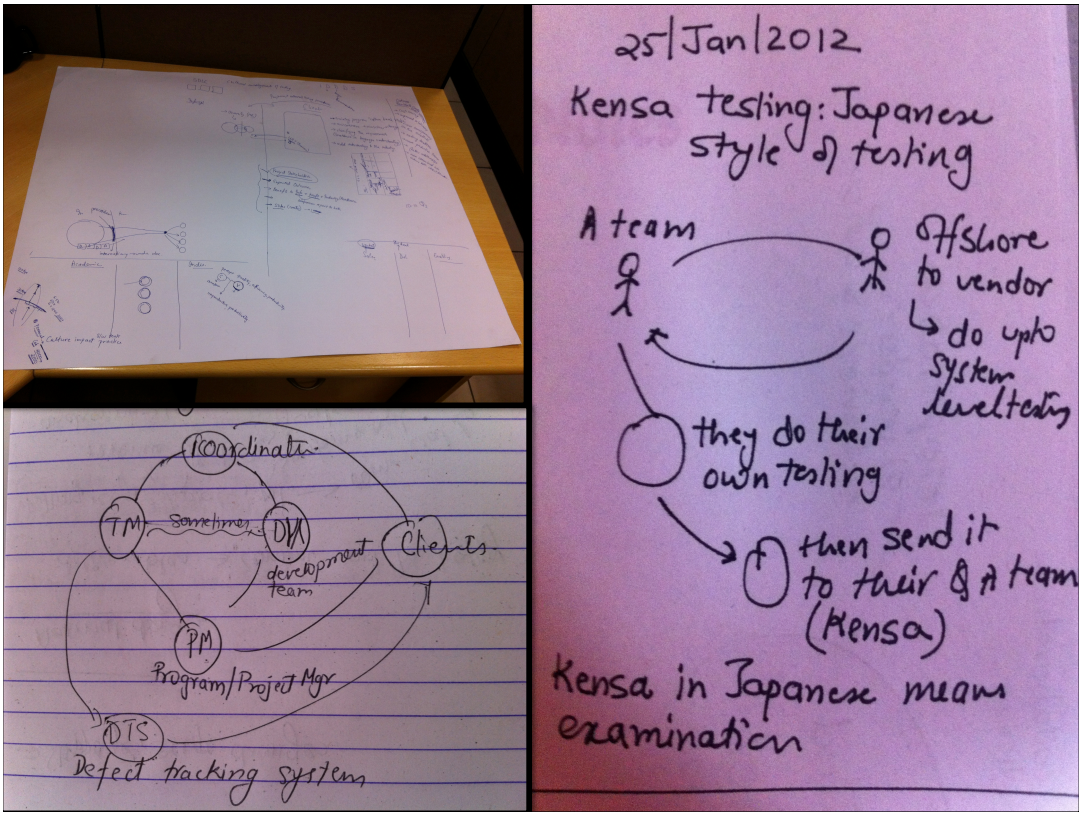


Figure 8: Handwritten Notes and Drawings Used for Member Validation.

and automation) of the client's software application.

For the U.S. Client (S3) Team, I had obtained prior permissions to interact with the clients. Fortunately, during the data collection period, some client members had visited India and so I had the opportunity to directly interact with them. In fact, the senior management (i.e., the software delivery management level managers in Figure5) of the U.S. Vendor (S3T1) Team formally introduced us to the U.S. Client (S3) Team when the client team visited the vendor organization in India.

Mexican Vendor (S3T2) Team. included members of the vendor organization working on the project for the Mexican Client (S3) Team. The partnership with this client was one and a half years old and during this period the vendor teams provided testing services to the client. The client was also from the retail domain (U.S. Client (S3) Team and Mexican Client (S3) Teambelong to the same parent company). Nine team

Table 3: Study S3 Participant Details

Team	Project Type	Testing Phase	Domain	Client Location	Number of Participants	Client Partnership Period (years)
U.S. Vendor (S3T1) Team	Testing	Planning	Retail	U.S.	5	3
Mexican Vendor (S3T2) Team	Testing	Execution	Retail	Mexico	9	1.5
Japanese Vendor (S3T3) Team	Development and Testing	Planning	Data-Center Systems	Japan	5	10

members directly participated and contributed to the study.

It was infeasible to get access to the Mexican Client (S3) Teambecause of the logistics reasons. Hence, I did not have the opportunity to interact with them directly. However, I was able to interact with a local (Mexican) vendor team member who was the onsite coordinator at the time of the study. Thus, I present the Mexican Client (S3) Team's perspective as reported by the Mexican Vendor (S3T2) Team.

Japanese Vendor (S3T3) Team. members of the vendor organization who worked on the project for the Japanese Client (S3) Team. At the time of the study, the client-vendor team pair were celebrating their 10th anniversary of partnership. The team was involved in test planning at that time to test the Data-center system of the clients. The Japanese Vendor (S3T3) Team was providing both the development and the testing services to the client. However, during our discussions, I concentrated more on understanding the Japanese team's testing activities. I had the opportunity to interact with 5 participants from the Japanese Vendor (S3T3) Team team.

I was informally introduced to Japanese Vendor (S3T3) Team members at a cultural-sensitivity workshop, where I was performing participant observations.[5] Because the interactions were not arranged by the Japanese Vendor (S3T3) Team's senior management, it was logistically difficult to get access to their client team. Thus, I present the Japanese Client (S3) Team's perspective as reported by the Japanese Vendor (S3T3) Team. All the participants who worked for the Japanese client had prior experience (at least two years) working with some U.S. client. Hence, they shared some interesting insights by comparing their experiences across different client groups.

All our participants had industry experience of at least three years and the senior-member participants (e.g., Delivery Head) had more than twenty years in software-testing related area. Neither of the projects was safety or mission critical (e.g., aviation, healthcare, or banking) but both were business critical to the vendor and the client organizations (e.g., the U.S. project was worth $1.5 million). The primary project-related responsibilities were documented in various official documents including the service-level agreements (SLAs), statement of work, process-definition documents, and test planning documents. Documents such as the process-definition documents included information related to the business domain knowledge, which was created and shared by the client team with the vendor team. Such documents provided high-level information related to test scenarios that the vendor teams were required to expand further to create test cases for execution. In additional, documents such as test planning documents were primarily initiated by the vendor teams, which included detailed test plans and test-executions steps in the form of test cases.

The *project-independent* participants included other members of the organization who were not directly associated with the GST practice. For instance, I interacted with members from the corporate marketing and sales team, which performs GST related market

[5]Cultural-sensitivity workshops are training sessions that the vendor organization conducted to bring cross-cultural awareness among their employees.

research to understand their competitors marketing strategies. For another instance, I also interacted with six members (internal and external to the organization) from the cultural-training group, which provides the cultural-awareness training at the organization. For yet another instance, I had insightful discussions with members from the Human Resources—to understand annual appraisal of employees and attrition information, Quality Assurance Team—to understand the client feedback information, and Research Division members—to understand their effort to improve the GST practice. Although because of time constraints, I was not able to analyze all this rich data that I collected from this participant category, whenever applicable, I used their data for validating my findings.

3.5 Summary

The table below shows the summary of the three studies that I conducted. The first column lists the various parameters. The next three columns show the values corresponding to those parameters for the three studies respectively. The row displays that the duration of the Study S1 was one month, Study S2 was 2 months, and Study S3 was four months. The next two rows list the number of interviews conducted and the respective hours of interviews conducted. The fourth, fifth, and sixth row lists the hours of observation, meeting, and information discussion related data that I collected. The seventh row shows the total hours of data collection, which is a sum of the total hours interviewed, observed, meetings conducted, and information discussions conducted. The eighth row presents the number of participants that I interacted with in each study. Finally the ninth row displays the number of locations visited in each of the studies.

As mentioned earlier, the previous two studies provided me with several opportunities to cumulatively learn, thus helping me to improve the execution of my final ethnographic study. Table 10 presents an overview of this cumulative learnings that I acquired. The learnings are categorized based on the aspect of the study that was affected: data access,

	Study S1	Study S2	Study S3
Duration of the Study	1 month (35 days)	2 months	4 months
Total Number of Interviews	32	29	47
Total Hours of Interview Data	21	29	32
Total Hours of Observation Data	not documented	3	12
Total Hours of Meeting Data	not documented	1	20
Total Hours of Informal Discussion Data	not documented	11	9
Total Hours of Data Collection	**21+**	**44**	**73**
Participants Count	32	23	44
Number of Locations Visited	2	2	2

Figure 9: Summary Table of the Data Collection of Three Studies.

physical area access, participant access, Institute Review Board (IRB) [6] protocol, or artifacts access. In addition, each learning is described based on its activity and presents the status in each of the three studies.

For instance, the first learning was a data-access related learning, where the issue was lack of permission to audio record the data-collection sessions (e.g., interviews and observations). Because I was not permitted to audio record the sessions in the first study (Study S1), I ensured to negotiate for this permission in advance before the remaining two studies started at the respective participant organizations. For another instance, the sixth learning was participant access related learning, where the issue was access to cultural training teams. In this case, I was unaware of the existence of such training teams in the first two studies. However, when I learnt that such teams exist at the beginning of the third study, I immediately started exploring opportunities to get connected to informants from the training group at the third organization. For yet another instance, the eighth category is IRB protocol related learning. In the initial two studies, I had designed the protocol such that I was expected to take informed consent of the participants, whose data will be used during the analysis. However, during the first two studies, I realized that this imposition of getting

[6] IRB is a committee that, according to federal law, protects the rights and welfare of research subjects.

informed consent from the participants was significantly hampering my ethnographic data-collection method. Because of this imposition, some participants were reluctant to sign any form (despite me explaining that this was an IRB consent form) in the corporate setting, although they expressed their willingness to share their experiences. Thus, acquiring consent had adversely affected (1) information gathering through informal sessions and (2) rapport building with my participants. Hence, in the third study, I modified the IRB protocol requesting the institute to grant me waiver for informed consent (refer to Appendix F). This modification was tremendously beneficial because it provided me the significant amount of flexibility in my data collection process.

In conclusion, every study done in the past, brought in new experiences that helped me improve my future studies, which reaffirms Spradley's note on ethnographic learning.

#	Category	Activity	Study S1 Status	Study S2 Status	Study S3 Status
1	Data Access	Permission to audio record	No	Yes (negotiated in advance)	Yes (negotiated in advance)
2		Permission to carry laptop to field site	No	Yes (negotiated in advance)	Yes (negotiated in advance)
3	Physical Area Access	Dedicated space available for researcher to sit besides the participants	No	Limited	Yes (negotiated in advance)
4	Participant Access	Access to onsite coordinators	No	No	Yes (negotiated in advance)
5		Access to client members	No	No	Yes (negotiated in advance)
6		Access to cultural training teams	(Unaware they existed)	(Unaware they existed)	Yes
7		Access to marketing and client-facing groop	(Unaware they existed)	(Unaware they existed)	Yes
8	IRB Protocol	Informed consent required from participants	Yes	Yes	Modified IRB to waive informed consent for better access to participants
9	Artifact Access	Access to artifacts such as project documents	Strictly prohibited	Available only to view from participant's system	Shared with the researcher

Figure 10: Table Summarizing the Cumulative Learnings from the Studies

CHAPTER IV

DESCRIBING THE CUSTOMARY PRACTICE OF GST

Cuff suggests that "customary actions, as they evolve, weave webs of meaning among a group of participants; these [meanings] form the very basis of culture according to some definitions" [40, pg. 5]. This idea forms the basis of this chapter, where I describe the customary activities that the practitioners (i.e., my participants) of Global Software Testing (GST) perform on a routine basis. These customary activities shape the foundational context that is important for understanding the cultural practice of GST.

Unless specified otherwise, the findings presented in this chapter are drawn from all my three studies—Study S1, Study S2, and Study S3— and they describe the global software testing as practiced by my participants from the three vendor organization—Vendor V1, Vendor V2, and Vendor V3. The chapter consists of various sections that describe the characteristics of GST experience (Section 4.1), the influence that the team structures have on the participant's experience (Section 4.2), and the challenges that they encounter while executing the GST-related activities (Section 4.3).

4.1 Characteristics of GST Experience

Considerably less research has been done on describing the rich experience embedded in the GST practice. Here, *experience* refers to the practical, valuable, and experiential aspects associated with testing and it involves the description of the practitioners' attitudes, behaviors, and emotions when they perform the testing activity. In this section, I concentrate on uncovering some rich experiences embodied in the practice. These experiences (1) contribute to the important process of creating the value of GST and (2) represent the meaning of the GST as internalized by its practitioners. Note that I report these experiences based on the GST practice that I observed. However, it may be the case that some

experiences may overlap with experiences encountered in the collocated testing practice settings too. Additional studies are required to perform systematic comparisons between the collocated and GST teams to make any conclusions.

4.1.1 Facilitating Training

One important experience that GST generated and supported was the training experience. This experience manifested in many different forms as described below.

Training to Becoming Good Developers. Junior participants' (those with two or fewer years of experience) attitudes toward testing was quite positive; they were generally enthusiastic and they strive to test well to ensure that the code going to the customer was of high quality. Participants strongly believed that testing helped them learn the system better so they could become good programmers in the future. Testing taught them the mistakes that they should not make when they start developing code. In Study S1, I found evidence of this kind of learning. In his spare time, test engineerS1T3.TM1 was building a sample graphical user interface application. The participant not only used his test cases to identify the functionalities of his application but also used them to ensure that he did not make the mistakes that he had found while testing the other applications. The participant mentioned

> ...then you think of all the aspects when you are testing. So testing, first, really helps you before I get into [the] development project and testing helps you in developing the code better.

Training Non-Computer Science Graduates. There is plenty of anecdotal evidence suggesting that in India it is common for non-computer science major students to eventually take up computer-science jobs because of the thriving Information Technology (I.T.) job market. Inevitably, some of my participants were also from non-computer science backgrounds. Particularly, my junior participants from non-computer science background found software development to be more challenging than testing because they were inexperienced in programming and lacked any educational training in the same. Additionally, they had

heard anecdotes from their friends about the difficulties in learning programming skills on job because of the demanding nature of the work under deadline pressures. Such anecdotes discouraged them from working on development projects as their first project: "but I am very happy that I directly did not get into a development project because my friends who have directly got into development say that it is very difficult for them to cope with the language learning as well as the deadlines" (S1T2.TM1). Instead, they preferred testing projects because they viewed testing as an opportunity to train and expose themselves to complex programming environments. Testing provided the necessary context for them to understand the code, system, and its complexities. Interestingly, participant S1T3.TM2 even considered herself to be "lucky" that she was put in testing and mentioned that she liked testing over development as it demanded more analytical thinking to break the code: "[I don't prefer development]...testing is more analytical thinking. I like to work on this kind of behavior (S1T3.TM2).

Testing as the First Project. I found that there was a preference in the industry to put junior (newly graduated) software engineers in testing projects because there were several benefits in having testing as a first project for them. For instance, it helped the junior software engineers to better understand the domain because testing provided a broad exposure of the software system being developed. For another instance,, as discussed earlier in this section, testing was considered to be helpful for learning good software-development skills. According to the junior participants, their seniors frequently showed them the benefits of being on testing projects during the early stages of their careers. For example, two participants mentioned that their senior had explained to them that being involved in testing would let them learn the product faster and in more depth. Once they have mastered the product, they would be able to make a stronger contribution to the development of the product. In fact, the senior had even given them examples of other seniors who were into testing earlier

and then had proved to be good developers. Although there were advantages of having testing as a first project for the junior software engineers, there are some challenges involved when less experienced engineers joined client projects. I discuss these challenges in a later Chapter (Section 6.4).

Making Users Familiar with the System. Not only did the vendor members use testing as a means to train their team members, but also the client teams used testing as a mechanism to train their users. Client manager S3.CO1 shared stories of the reluctance that some of her colleagues expressed towards outsourcing and offshoring the testing activities because they were afraid of losing the opportunity to train their team members (and users): **(S3.CO1)**

The biggest concern I hear from my clients is around the business as far as they go. Some of them use the testing to do training. So they are afraid, if I use a testing team they are going to lose that opportunity . . . I'm going to lose the opportunity to get my business up to speed. So by bringing in my key users during integration [testing] you know, that's how I'm actually training them. (Quote 1)

Learning from the Clients. In Study S3, I found that the experience of working on the projects with clients from different organizations and cultural groups provided participants with unique opportunities to learn new strategies and techniques to perform testing activities better. In particular, the participants working for the Japanese Client (S3) Team were impressed with their client's approach towards testing and their style of conducting testing activities (refer to Section 5.1 for detailed discussion), which they acknowledged and greatly appreciated. The Japanese Vendor (S3T3) Team members initially were of the opinion that they had performed thorough testing. However, when the software system went to the Japanese Client (S3) Team's testing (referred to as "Kensa" or quality assurance) group, this team reported many defects. The Japanese Vendor (S3T3) Team was astonished at the thought process the client's testing team adopted to test the system. After this experience the Japanese Vendor (S3T3) Team members started learning from and adopting to a similar

thought process to create advanced test scenarios that better covered the testing scope of the system. This is expressed by a technical management member from team S3T3:

(S3T3.TM1)
Now I'm not in a position to say that they have an advanced tests because having gone through that myself, I feel that this is what the product should also be tested for. But at that point of time, we definitely did not test each of these scenarios. And we still thought that we had sufficiently tested it and we did ship it to Kensa [Japanese client's testing team]. But then some of these things did not pass because obviously the code was not handling. Looking at the requirements document we did not anticipate that they will test in such a situation also. (Quote 2)

Thus, cultural exposures and the experience of working with diverse client teams benefited the vendor team to train themselves better at the testing tasks.

4.1.2 Selecting Manual or Automation Testing

In the previous section (Section 4.1.1), I reported various ways in which members of the GST teams—both the client and vendor teams—used testing as a means to train their employees. This training is mostly done using manual-testing approach, where the test engineer executes the steps of the test cases manually instead of using an automated program to execute the test cases. My research suggests that this is one reasons why, even in current times, the vendor organizations always conduct the first cycle of testing manually because the manual testing brings in this important experience of training into the GST practice and helps them judge which aspects of the test cases suite will be worth automating.

Not only does manual testing facilitate this rich training experience, but also manual testing appears to exhibit more reliability in the testing process than the counter method of automation testing—where scripts are programmed that can be run automatically to execute the test cases. This lack of trust in the automation testing method was evident in Study S2 where the participants shared the reluctance of their client to adopt the automation testing approach for their project [119]. The reason for this reluctance was that the client seemed skeptical about the process of using one computer program to test another program and, therefore, doubted the reliability of test automation.

(S2T1.TM1)
So onshore people were thinking that manual testing is more reliable because you [test engineers] can actually go and execute the test manually instead of relying on a tool. Because when you click on the button [in the automation tool that executes the automation script], the tool will do everything itself. But, that [tool] is coded by someone so then they [clients] were thinking that the reliability is not that much... (Quote 3)

Moreover, despite the benefits of automation (i.e., predictable, repeatable, and efficient test executions), it was often not accorded high priority in the GST setup. Automation had low priority because of its large up-front investment in training automation test engineers, creating the automation environment (buying the necessary test automation tools), and investing considerable amount of human effort in automating manual test cases. Consequently, not all test cases were automated. Automating test cases was preferred only when it was certain that the test cases would be executed several times as part of regression testing, high precision was required (e.g., comparing each pixel on the screen), or running the test cases was too time-consuming and repetitive (e.g., load testing for 100 users). Moreover, determining which test cases to automate was usually performed by the manual test engineers or clients, not the automation test engineers.

The automation test engineers among the participants reported experiences in which they were expected to create a fixed number of automated scripts each day, irrespective of the complexity of the scripts, the stability of the code whose test cases needed to be automated, the incomplete knowledge of the test engineer about the scenarios to be scripted, or the skill level of the test engineer. A few participants from Study S2 also commented that they were not given all the test cases so that they could understand commonality among the scripts and modularize the scripts to ease maintenance. Instead, they were asked to automate the scripts on an ongoing basis. The participants thought that this was an inefficient practice because it involved much rework on their part, which was not considered as actual work by the managers and the clients. Nonetheless, this practice was followed because the clients wanted a pre-determined number of scripts automated by the end of each day. Thus, the number of scripts automated per day was a critical measuring component of

productivity.

Despite these challenges in adopting automation testing approach in the practice, the vendor-side test engineers were keen to work on automation projects. The reason was that the process of automation involved programming test scripts. This programming process helped the participants enhance their programming experience, which was important for them to move to software-development work profiles in the future. Most of the vendor-team members were keen on moving to development projects as soon as possible because of the socio-cultural status associated with being a test engineer (e.g., Section 2.5.3 discussion on testing as second class citizen).

4.1.3 Bringing in Power

Despite its monotonous nature, an important factor contributed to the enthusiasm the participants exhibited towards testing. This factor was the *feeling of power* that the testing profile manifested. When participants had responsibilities of testing modules but they also were given the ownership of those modules, they possessed the feeling of power to control testing-related decisions. This power was in the form of deciding the order in which they would test the modules, interacting directly with the development team for those modules, managing other members testing the same module, and/or most importantly deciding whether the product can be shipped (i.e., "signed-off") in the upcoming release based on its current quality. This power arose particularly in configurations where the testing and development teams were two separate entities. When there was no overlap in the members between the two teams (i.e., when resources were not shared across the two teams), the test engineers' decisions were not influenced by developers actions and constraints.

This feeling of power brought in motivation and enthusiasm among the test engineers. For instance, I found that sometimes the participants were allocated a few modules for testing that they were said to *own*; they were responsible for testing each of the different versions of those modules. Although testing the different versions seemed monotonous

to many of these participants, the participants were enthusiastic to test them because they enjoyed the ownership of those modules. This ownership was giving them the power they appreciated and enjoyed. Thus, this feeling of ownership seemed to be a crucial factor that contributed to their enthusiasm. In fact, one participant (S1T3.TM2) shared an incident that supports this finding. She mentioned that although she was bored with testing the module M (renamed) repeatedly, she was the one who knew the most about module M, so ther managers would listen to what she had to say.

(S1T3.TM2)
Currently I am the only one who knows ... about [M] module, so [manager] does not say a lot to me. He has to listen to what I have to say. (Quote 4)

Many participants also mentioned that they liked the feeling of not being questioned and having the authority to question others (e.g., development team members) when something went wrong. They enjoyed the privilege that, without their approval, no application could go into production. One participant stated

(S1T3.TM3)
I am always there to ask a developer and I don't have to answer anybody. (Quote 5)

This finding supports the idea that there exists a relationship between responsibility and power. When participants had responsibilities along with power, the enthusiasm of the participants was greater than when they had only responsibility and no power. Some participants had both (1) the responsibility of getting their tasks done and (2) the power to make decisions regarding their tasks, to directly communicate with the clients, and to work in their own way. These participants were quite enthusiastic about their work. However, some participants had only responsibilities and no power to make decisions; the power was reserved for their senior managers. These participants were not as enthusiastic as the former. Instead, they expressed the frustrations because of the lack of power to make the relevant decisions.

It appears that the responsibility and power relationship seems to be dependent on the hierarchical structuring of the teams. The more the team was structured, the more the distribution of responsibility and the power among the members of the team was separated.

For example, if a team has many hierarchical levels in the project, such as project leads, group leads, and module leads, (as was the case in all my studies) there seemed a higher possibility that responsibilities and powers will be given to different people. The members higher up in the hierarchy would get the power whereas the members at the lower levels would get the responsibilities. This indicates that a hierarchical structuring might influence the enthusiasm in the teams. However, current data are too limited to draw any strong conclusions. Hence, additional studies, to investigate this thread further, can benefit to better understand the relationship between team structuring and enthusiasm.

4.1.4 Capturing Screenshots as Proofs

When software-development phases such as design and coding phases are performed, they produce software artifacts such as the design documents and the code as an end product. However, when testing is conducted, the end product is just a status indicating whether the test cases passed or failed. Because of the lack of any artifacts as an end product, it is difficult to prove whether a particular test cases was indeed executed or the status was just faked. This concern is reported as one of the common concerns in the GST setting, where testing is being done by vendor teams at offshore. Because there is no concrete proof that the test case was in fact executed, there was one procedure adopted by GST practitioners to address this concern— the procedure was to capture screenshots of every step in the test cases execution process. The practice has been to capture the screenshots and upload them at a common repository location, where the clients can access them whenever they desire. The screenshots provide the required proof of test-cases execution. Interestingly, this screenshot procedure was sometimes even specified in the compliance standards defined between the client and the vendor organizations and it differed from client to client. for instance, participant S3T2.PM2mentioned his experience working with one client in the past who expected them to “take printout of the screen shot and . . . physical[ly provide] signature.” Thus, these screenshots play an implicitly important role in building the credibility of the

vendor teams (I discuss this relation in detail in the Section 6.2).

The participants from Study S2 and Study S3 reported that this screenshot taking procedure was a tedious and time consuming process. Nonetheless, the participants found this mandated procedure to be useful in many ways. First, during the regression-testing cycle, the test engineers were able to refer to these screenshots to recollect the steps of execution. Second, for lengthy test cases (i.e., which had many steps), the screenshots provided a mechanism to verify whether the test-cases steps were executed correctly, when investigating the reason behind failure of a test case. Third, it generated artifacts and created a reference system that helped answer client queries, which were directed on the past executions. Lastly, the screenshots became the software artifacts for the testing phase and provided the proof of testing service:

(S3T2.PM3)
Screenshots provide the proof of testing services and input for business team to approve critical business transactions. We execute test case with certain master data, which we agreed with customer before start of execution cycle. All testing is being done remotely and requires validation/approval from the business people (i.e., the client team). In order to facilitate this and ensuring team members at remote location are executing correct test cases with pre-defined process and master data, we need to capture, screen shots in [XX] tool. (Quote 6)

4.1.5 Encountering Quality-Dilemma Situation

The test engineers often reported that they experienced more pressure than the other teams involved in software development (e.g., development teams and design teams) because they must accommodate and adapt to delays in the design and development phases. The participants mentioned that delays in the design and development phases occur frequently in the global (i.e., outsourcing, offshoring) setting and, hence, it is commonly accepted norm in the practice. Consequently, the test engineers' time to perform and finish testing related tasks is significantly reduced and they are expected to perform their testing tasks in the reduced duration of time.

These conditions have led to the situations where the test engineers struggled to decide whether they wanted to finish the assigned testing tasks on time by potentially compromising the quality or maintain the quality at the cost of possibly not finishing the tasks (despite working overtime). I refer to such situations as the *quality-dilemma situation* [119]. Thus, the test engineers face a challenge in ensuring that high-quality testing is done under strict time deadlines (some challenges are discussed in detail in Section 4.3.1). In other words, high personal values among the test engineers appear to be crucial for ensuring high-quality testing under time pressure.

The quality of work in testing directly relates to the quality of the product because testing is the activity that facilitates the creation of a good-quality product by uncovering bugs in the software that can be fixed before release. I found that the test engineers strived to avoid quality compromises by adopting some approaches that can help ensure that the tasks are completed without jeopardizing the quality. For example, participants from Study S2 reported that they tried to alert their managers in advance about such situations so that the managers could take appropriate actions. At times, the managers were indeed receptive to such situations and took appropriate measures to resolve the problem, whereas at other times, the managers ignored the testers' pleas for additional time to perform the testing.

The study analysis suggest that such quality-dilemma situations are influenced by several factors such as giving excessive importance to numbers, ignoring task complexity, and long and rigid communication structures. Excessive importance to numbers was evident when participants (Study S2) hinted that their productivity was typically measured in terms of the number of test cases created, automated, and executed each day.[1] The problem with such (implied) productivity measures was that the client teams seemed to neglect the complexity of the testing related tasks and situations. One participant (S2T1.TM1) mentioned that because his onshore managers' main focus was on numbers, the managers were less

[1]Participants did not explicitly mention that these were the productivity measures. However, they implied it and hence this is my interpretation based on my analysis.

bothered about the task complexity.

(S2T1.TM1)

His [onshore manager's] main focus was on numbers, so we were completing four scripts per day, three scripts per day [i.e., creating scripts for automating test cases]. They [the client team] are [the] least bothered about the complexity. And that was the challenge that we had to manage. (Quote 7)

Moreover, the long and rigid communication structures because of the deep hierarchical cultural systems of these Indian vendor organizations created additional challenges. For instance, some participants mentioned that sometime to address the quality-dilemma situations they wanted to escalate the situation to the respective decision-makers (i.e., higher management or clients). However, there were two main challenges with escalating. First, under time constraints (when the release dates are close), the managers were less receptive to genuine problems of the test engineers because they in turn were answerable to the clients. Second, the deep hierarchical organization structures at the vendor side resulted in longer time to get an approval or raise an issue because the approval had to go through several levels, each level adding to the delay. Ironically, the time was the scarce resource in such situations and hence escalating the situations sometimes did not help. Unfortunately, the participants were left with the feeling that they were working on a project that was bound to fail; they were left to live a bad experience.

When complexity is ignored and pressure is high, the chances for mistakes increases and thus, there are compromises to the quality of work resulting in (unintended) compromises to product quality. The participants shared their experiences when they encountered such dilemma situations, which resulted in such quality compromises. For instance, one participant from Study S2 (S2T1.TM3) mentioned that sometimes he was forced to skip some test cases or test-case execution related steps because of the lack of sufficient time. For another instance, in a similar other situation of lack of time, one team's members in Study S3 omitted the steps of taking screen-shots and uploading the proof of testing (refer to Section 4.1.4). This had led the client team to believe that the vendor team did not actually perform the testing at the offshore site, contributing to the factor that caused the

escalations (described in detail in the Section 5.2) and the loss of client's trust (Section 6.2).

For yet another instance, participant S2T1.TM2 mentioned that

(S2T1.TM2)

I want to add one point. We always get less time than decided earlier, hence putting extra pressure on the testing team as we have promised some delivery date. In case of too much pressure, it always affects your quality. If I am being forced to execute more test cases, quality decreases automatically as productivity will not be good outside working hours. This increases mistakes too...they [testers] may skip some parts. We've to keep evidences of everything we execute[??], sometimes however to hurry they skip evidences thing, where it is taking a lot of time, thinking nobody is going to review the evidences. Sometimes they can skip some comparisons if that is taking a lot of time...But, the notepad [where the test results are stored], you can edit it very easily. Some mismatching values can easily be changed. So you have to prove that you have done good testing and your evidences are matching the query result. But this is where we can compromise the quality of work.

(Quote 8)

Thus, these quality-dilemma situations were common and seemed to occur quite frequently in the GST practice.

4.1.6 Governing Factor: Time

Rooksby reports that time is a significant factor in determining the organization of testing-related activities [113]. Interestingly, for some clients (e.g., Mexican Client (S3) Team) "time was even more important than quality" (mentioned by S3.OC1). I also found that time pressure—manifested as insufficient time to perform the allocated tasks—plays a significant role during the test-execution phase. Often, the factors that contribute to creating pressure either reduce the time available for accomplishing the allocated tasks or increase the tasks to be accomplished within the allocated time. In either case, the tasks to be completed by a tester for each unit of time increases.

The participants shared many stories about their experiences in dealing with time pressures during this phase. Different participants perceived pressure situations differently. For some participants, such situations occurred when managers and onshore members queried them frequently about the status of their testing tasks. For other participants, pressure situations occurred when they realized that they would not be able to finish their tasks in the

given time. For yet other participants, it was the time when they had "signed off" on the code, and nothing could be done even if they found a bug that was not caught by the testing teams. This situation of experiencing pressure after the test-execution phase is over (i.e., in the client-management phase) caused anxiety among the test engineers. Moreover, pressure was perceived as bad at times and good at other times. Some participants mentioned that they did not appreciate unnecessary pressure as it demotivated them. However, they viewed pressure situations as a way to measure their capability of performing tasks in a short period of time. One participant stated

(S2T1.TM1)

They [the managers] will still put pressure on us, because sometimes you [test engineers] do have the capability of doing the thing [performing the task] when you are in pressure [under pressure]... you will screw [skip] your lunch and tea breaks and get the work done. So in that case, putting the pressure is okay. It might happen that he [the manager] may put pressure [on the test engineers] straight for two days, and thus, you may be able to do three days tasks in two days rather than four. So for the manager, it is finding the right time when to put pressure and when not to. It's not always that the manager has to put pressure on the team, it's sometimes that he has to communicate to the higher manager and extend the deadlines. (Quote 9)

Most participants reported that the testing team was the one that has the most stress of all the teams involved in the development of the software because of the immense time pressures during the test-execution phase. The participants mentioned that they experience such pressure during almost every release. The most common factor that contributed to the creation of the time pressure was delay in code delivery from the developers, which corroborates the findings of Rooksby and colleagues [113]. One participant expressed this experience

(S2T2.TM2)

A developer might take 20 days to develop [code] but [test engineers] have to test it within the time [allocated] and within the given deadline. . . . Developers can [miss] the deadline; [test engineers] cannot [miss] the deadline. [Meeting the deadline] is the major challenge as a tester. If there are 10 days for the delivery, and [the] developer was assigned five but he takes eight, . . . it is [the test engineer's] time that is being cut not [the developer's]. (Quote 10)

Other factors causing deadline pressures that the participants mentioned include

- Incorrect time estimation for testing-related activities, which results in the allocation of insufficient time for the planned activities
- Sudden shortage of resources (e.g., because of unplanned leave), which increases the burden on test engineers
- Impromptu requests for high-priority activities, which distracts test engineers from the scheduled activities
- Preponement of the deadline (this occurred rarely in the participants' experience), which decreases the time available for completing the scheduled tasks
- Slow defect fixing, which, in turn, results in delays in retesting after a defect fix
- Technical problems, such as data-setup failure, system failure, or issues with setting up the testing environment, which cause delays in completing the planned testing activities

As a consequence of the time pressure, the test engineers were expected to work extra hours to finish the scheduled tasks. They were asked to stay late or work on weekends. Moreover, some participants mentioned that the practice of performing testing activities in less time than ideally required was not a one-time occurrence, but a recurring phenomenon.

4.1.7 Striving for Motivation

The nature of the testing job is peculiar because of the unique characteristics it possesses. First, testing jobs seem to have the characteristic of "generating more work on doing work." For example, if a test engineer finds bugs in the code, she is inviting herself to work overtime because once she reports the bugs, she needs to retest them after they are fixed. Retesting is an official expectation because it is explicitly mentioned in the service level agreement documents: "Retest the defect onces fixed and update the test script accordingly" (from the Service Level Agreement (SLA) document of S3T1 team). Thus, under pressure, when the test engineers are already stressed to finish their work in a limited time, this property of testing job of increasing their work creates additional challenges. To add to their

struggles, as mentioned in Section 4.1.6, many a times there are delays in code fixing from the development teams, which result in additional delays. However, such challenges seem to have never been acknowledged by the managers and clients. Second, it is commonly believed that test engineers are considered to be second-class citizens (I discuss this theme in detail in Section 2.5.3) [113, 117, 118]. Hence, their profile seem to be viewed as less important than other profiles (e.g., development), which is another demotivating factor. Although this perception appears to be changing gradually (as discussed in Reference [118]), testing is still strogly perceived as a low-status work profile in cultural societies such as Indian society. Third, exhaustive testing is not feasible because there is no good way to decide an optimum measure, which if met, will show that the test engineer has done a good job at testing. This property of testing seems to result in test engineers being less frequently appreciated (by both clients and vendors) for their efforts, which leads to reduced motivation among the test engineers.

Thus, maintaining high-motivation levels for test engineers becomes particularly challenging because of these three main reasons. Because the test engineers are responsible for assessing and ensuring that a good quality product is being built, it is important to ensure that their quality of work is noticed and appreciated, not only by the client organization but also by the vendor organization.

4.2 Team Structure's Influence on Testing Experience

The team structures, within the vendor organization or client-vendor team, had significant impact of the kind of experiences that the participants encountered. In this section, I discuss how the client-vendor team structure influenced the pressure experienced by the test engineers (Section 4.2.1) and the vendor-organization's team structure caused increased frustration among the test engineers (Section 4.2.2).

4.2.1 Client-Vendor Team Structure: Increased Pressure

As I discussed in Section 4.1.6, the participants reported that, in general, the testing teams experienced considerably higher pressure than other teams, such as design and development teams, as deadlines approach because testing is the last phase before version releases occur. However, I found an interesting dynamics in play between the pressure experiences and the team's structures.

In Study S2, U.K. Vendor (S2T1) Team and U.K. Vendor (S2T2) Team reported a greater pressure and stress level than U.S. Vendor (S2T3) Team. On investigating this thread further, I found that the inter-organizational (i.e., the onshore-offshore) team structuring played a role in this difference of experiences. On the one hand, U.S. Vendor (S2T3) Team directly reported to the clients (i.e., without the presence of any intermediate onshore team composed of local hires). Nonetheless, some of their offshore team members were deployed onsite to work closely with the client team as onsite coordinators. On the other hand, for U.K. Vendor (S2T1) Team and U.K. Vendor (S2T2) Team, the vendor organization had an onshore office where the local hires formed the onshore vendor team and they played the role of the onsite coordinators (refer to the Section 3.3.2 that discusses the team structures). This onshore team worked closely with the clients and played the role of the intermediary between the client and vendor teams. This intermediary team, composed of the U.K.-based local hires, created additional struggles for the test engineers.

Because of the presence of these U.K.-based local hires as the onsite coordinators, the clients lacked clear visibility of issues, such as internal delays created by the offshore development team in delivering the code to the testing team. The participants from U.K. Vendor (S2T1) Team and U.K. Vendor (S2T2) Team mentioned that, despite delays in receiving the deliverables from the design and development teams, the major release dates did not usually change. This adherence to the original schedule increased pressure on the testing teams to accomplish the pre-committed number of tasks, but within a time period that was less than originally estimated. The participants from U.S. Vendor (S2T3) Team

mentioned that the majority of the time, their clients were aware of such internal delays and, hence, the testing team had the opportunity to take adequate steps (e.g., re-negotiate deadlines, request more resources, or reduce the scope of current release) to address the issues that arise from internal delays.

Interestingly, in Study S3, the Mexican Vendor (S3T2) Team had a client-vendor team structure similar to the U.K. Vendor (S2T1) Team and U.K. Vendor (S2T2) Team: they also had an onshore team composed of Mexico-based local hires. The Mexican Vendor (S3T2) Team experienced a different set of challenges. For this team, the Mexican onshore team, who played the role of onsite coordinators, were rigid about their work hours and working styles. They were not willing to work for beyond work hours, which was a challenges for this onshore-offshore model. Moreover, factors such as the language difference caused additional challenges. As a result, the offshore team reported that these struggles made it difficult for them to meet the deadlines. In fact, just before I conducted the study, the team had missed a deadline, which led to a serious escalation to higher management. I discuss this escalation situation in detail in Section 5.2.

The episodes in these two separate studies reinforce the idea that the team structures enact as controlling agents and they influence the experiences lived by the GST practitioners.

4.2.2 Vendor-Side Team Structure: "Multiple Channels of Hierarchy"

In all the three organizations that I studied, the organizational structures included deep hierarchies, resulting in long and rigid communication channels. As mentioned in Section 4.1.5, during the test-execution phase in the GST setting, when the deadline pressure peaks, these long and rigid communication chains posed significant challenges. The test engineers mentioned that the challenge was especially magnified when, faced with issues, they did not have the authority to make decisions. In these situations, they would report

the issues to their managers. However, often, the managers would defer the decision making to the clients. The participants reported that they were unsure whether the managers were reluctant, or not in a position, to make decisions. Furthermore, the participants sometimes doubted whether the managers communicated the issues to the clients, or whether they conveyed the issues but that the clients were unwilling to take any action. The participants believed that, often, the managers were hesitant to refuse the client's requests despite knowing that the requests were difficult to satisfy (I discuss this behavior in detail in the Section 6.1). One of the test engineer (S2T3.TM2), performing automation testing, shared her complaint about how her team was expected to work inefficiently and automate at least 10 test cases each day despite the code being unstable (code stability is crucial for successfully automating test cases and avoiding rework).

(S2T3.TM2)

Interviewer: ...where do you see the issue in all this?

Participant: Whether the clients are like that or the manager[s] is[are] like that I don't know. It's also like, I don't know where the things are getting masked. As I cannot directly talk to the client, it's difficult to find out.

Interviewer: Have you ever tried to talk to the clients directly?

Participant: It's like others can talk [to the clients], right? If I can talk directly, others who currently have access to the clients can also talk and discuss these issues currently... See, its not like I are not reaching [meeting] the deadlines or I lack the information [and the] skills or I are not punctual. It's like I have some practical concerns and I are not understanding [why] the clients is not understanding these practical things.

(Quote 11)

Because "there are multiple channels of hierarchy," conveying the issues to the clients was often delayed and as a result the issues reached the clients close to deadlines. Consequently, the issues were frequently ignored or remained unresolved. This, in turn, resulted in frustrations among the test engineers because they believed that the issues were bound to resurface in the future, leading to them to relive the bad experiences and frustrations.

4.3 Challenges Encountered and Strategies Adopted

In this section, I describe the impediments to conducting high-quality GST under time pressure. I present the general challenges faced by both manual and automation test engineers (Section 4.3.1). Also, I discuss some strategies that the teams adopted to overcome some of these challenges (Section 4.3.2).

4.3.1 Challenges of High-Quality GST Under Pressure

A pressure situation increases the likelihood that the quality of testing will be compromised (Section 4.1.5 and Section 4.1.6. To meet the testing goals and ensure that the quality of testing is not decreased, the test engineers must expend extra effort. my study revealed that their tasks can be made even more difficult by some complicating factors.

Information gaps. Lack of sufficient information is a common problem in the GST setup (e.g., as observed by Damian and colleagues [43]). However, under pressure conditions, lack of sufficient information further aggravates the situation for the test engineers. Most participants mentioned that they have trouble obtaining information pertinent to their testing tasks. For example, they have difficulty obtaining information about how a bug is reproduced, how a bug is fixed (so that the fix can be tested appropriately), and what priority is set for testing a module. Finding the appropriate contact person (e.g., someone involved in fixing a bug) is also difficult at times. The participants reported instances in which they did not know whether some crucial documents were updated by their client/onshore team members. One of the participants (S2T3.TM1) mentioned
(S2T3.TM1)

This is again a gap. While putting in the data, they [developers] should write out that this is the fix and [if] any clarification [is] required, please contact this particular person. What I did [was] that out of 10-20 documents, I tried to figure out who fixed this particular problem, then I contacted that person. That person was not available at that particular time, and I was told that I had to finish it by 3 o'clock so that by 3 o'clock, it can go into production. Now, 4-5 hours went only in finding out how to get that particular information so that I can verify what I have tested is correct or my approach is ok or I needed some more information. (Quote 12)

Even in cases where the appropriate contact person is known, obtaining information can still be challenging. Some participants mentioned incidents in which the onshore team members expressed resentment (e.g., by closing the chat window) when participants queried them frequently for information. S2T3.TM1 stated that

(S2T3.TM1)

The onshore and offshore communication is not happening as it should. So I have experienced myself that onshore people, most of the onshore people not all, there are some who are willing to help you [but the majority of the others are not], they are not free to help you always. So management should now work in building [a] better relationship between onshore and offshore developers, designers, managers, and testers whatever is their area of work. Sometimes they will even be rude to you: you ping them and they will close the chat window...It happened with me. I pinged an on-site developer and he simply closed the chat window. (Quote 13)

Visibility issues The visibility issues relate to clarity and understanding of the testing activities that are performed by the testing teams and that are expected by the client teams. Lack of visibility seems to be a significant hindrance to the smooth functioning of the testing activities in the teams I studied. This lack of visibility appears to manifest in both the testing and the client teams: the testing teams lack visibility into the expected testing activities, and the client teams lack visibility into the issues encountered by the testing teams while performing those activities (often resulting in the issues not being addressed). For example, the participants described situations in which the development teams were fixing high-severity bugs (i.e., critical bugs), which needed to be retested in a short period

of time by the test engineers, but the test engineers lacked visibility into the types of bugs and the types of fixes being performed. The participants also reported situations in which the issues encountered by the testing teams, such as data-setup problems, delayed delivery of code by the development team, and lack of required information, were not made visible to the clients.

Lack of acknowledgment of pressure situations One of the common complaints among the participants was that the senior managers or clients often do not acknowledge the pressure situations under which test engineers perform their tasks. Instead, the senior managers or clients conclude (unfairly) that the test engineers' quality of work is poor. The participants mentioned that, at times, they are asked to do unreasonable testing activities. For example, they are asked to test and release a quick fix (instead of addressing a problem thoroughly) to ensure that service-level agreements are not breached, or start testing even when the code is not ready to be delivered to them.

Clashing or misunderstood priorities In a pressure situation, it is crucial for the participants to understand how they should prioritize their activities. The participants reported experiences in which they were unclear whether an issue was high priority and, if so, what needed to be done. Also, they mentioned that, at times, their priorities clashed with the priorities of the developers. In such cases, they did not get the desired support and help from the developers to enable them to perform their tasks on time.

4.3.2 Strategies Adopted to Overcome Some Challenges

To mitigate the pressure situations, test engineers adopt different strategies. The common strategies include requesting additional resources, starting different activities simultaneously, and prioritizing testing tasks (based on developer reputation, manager's expectations, and the criticality of system modules). In addition, there were four interesting strategies that the participants reported that they use to cope with the pressures in the test-execution

phase. Some of these strategies are not peculiar to the GST setting—they may also be practiced in collocated (in-house) testing.

Rapport building with clients. One participant (S2T2.TM1) said that he expended extra effort in building and keeping good communication relationships with the clients and onshore members. This let him reduce information gaps and communication breakdowns, which occur in pressure situations. Moreover, the establishment of a good rapport helped him obtain relevant information quickly under time pressure. The participant mentioned

(S2T2.TM1)

First and foremost is effective communication. More and more . . . contacts that you [test engineers] have [it] help[s] you a lot, because the more contacts you have, its easier to get the information and you can get more . . . information actually when you have more . . . people to whom you can contact. So this [having more contacts] is one key point... [It is important to know] how to actually maintain your communication [possess good communication skills]...because at times if you bug people more they [client team members] get irritated, they stop talking to you, they stop responding, so you need to have that skill that you keep asking questions even then the person is not bugged with your questions. So that is one thing... I have observed. It is very much required especially in testing, because when you are testing something, you are actually involved in the business knowledge. [You are interested in] what business is actually going on...So when you need to do the business, you need to talk to designers who are actually very busy people. They hardly have time to talk, so you need to have good communication tactics [with them]. (Quote 14)

Announcing risks in advance Another participant (S2T2.TM2) mentioned that, when she had insufficient information, she would add notes to the test-case documents. These notes were to inform others about the lack of complete information and that she had performed the tasks based on the limited knowledge she had. She said that this practice helped her make the clients and managers aware of the situation under which she was performing her testing tasks.

Scheduling a mix of difficult and easy tasks The participants mentioned the strategy of interleaving difficult tasks with easy ones to ensure that they met the expected task-completion rates. For instance, one participant reported an experience in which he was expected to automate three manual test cases each day, irrespective of the complexity of the test cases. Initially, he tried to explain to the client that the complexity of the test cases varied significantly and, therefore, a rigid and fixed completion rate was unreasonable. However, the participants mentioned that because the pressure on the project was so high, the clients did not consider or appreciate the challenges faced in handling complex test cases. Thus, the participants adopted the strategy of mixing the creation of difficult and easy scripts so that he could meet the desired automation rate.

Providing support to speed up defect resolution The test engineers got more time to understand the functional requirements than the developers because the test-preparation phase overlapped with the developers' coding phase. Thus, during defect fixing, they extended support to the developers by being information providers and helping the developers understand the functional requirements. Additionally, one participant mentioned that occasionally she mediated to help the development team members resolve the ownership of a bug (i.e., who would be assigned to fix the bug.) The participant mentioned that, in some cases, she expended extra effort to consult the design team, understand the potential root cause of the bug, and accordingly, inform the respective development team member to take ownership of the bug and fix it.

CHAPTER V

DESCRIBING THE CULTURAL PRACTICES IN GST

Culture is deeply rooted in individuals and their everyday practices because culture molds the way people think [102, 124]. Thus, it is no surprise that the everyday practice of software development is also greatly influenced by culture (e.g., [50, 120, 118]). With the growing globalization of jobs, software teams typically have an inter-cultural group of members, teams distributed across different geographic locations, or both.

In this chapter, I describe the rich cultural experiences embodied in the GST practice that I observed in Study S3. This chapter discusses findings primarily from Study S3 because that was the study that significantly concentrated on investigating cultural influence of the GST practice. The chapter covers accounts from two separate pairs of teams studied in Study S3. First, Section 5.1 presents a comparison of GST practices as adopted by the clients—U.S. Client (S3) Team and Japanese Client (S3) Team. Because I did not have the opportunity to conduct extensive client-side studies (refer to Section 3) due to approval and logistics constraints, the accounts shared in Section 5.1 are largely drawn from the vendor participants' perspectives narrating their client experiences. However, based on discussion with the vendor teams about their respective clients' practices, there was rich comparison that emerged and I report on the findings from this comparison in the first Section. Second, Section 5.2 provides a client-vendor experience that illustrates how cultural clashes resulted in an escalation situation. I describe the situation and discuss the participants' explanations for the cause of the escalation.

5.1 Comparing Vendor Perceptions of Clients' Testing Practices

Based on the discussions with the vendor team members of the Japan and U.S. team, some interesting insights about their perception of the clients' cultural practices emerged from the

Study S3. I discuss these insights in the form of a comparison of software-testing activities of these two teams from the vendor teams' perspectives. As described in Section 3, the Japanese Client (S3) Team had a 10 years partnership with the Vendor V3 whereas the U.S. Client (S3) Team had a three years partnership with them. Despite the long term partnership, the Japanese Vendor (S3T3) Team faced various challenges (including lack of trust) working with a Japanese client as compared to working with a U.S. client; I illustrate and discuss these challenges in this section. This comparison was mainly possible because all my Japanese Vendor (S3T3) Team members had considerable (atleast two years) prior experience working with both U.S. and Japanese clients.

Also, the Japanese Vendor (S3T3) Team team was providing both development and testing service to the Japanese Client (S3) Team whereas U.S. Vendor (S3T1) Team was providing only testing service to the U.S. Client (S3) Team. Despite the differences in the services being provided by the two vendor teams, which may be leading to different experiences, the cultural influences on the GST practice are still comparable because, according to my Japanese Vendor (S3T3) Team participants, testing is the phase when the cultural (challenges) differences manifest the most: "one thing which we have learned from Japanese Client (S3) Team is the way they do software testing . . . the [cultural] challenges that we experience [pauses to think] . . . is manifested when they do their testing."

This section describes my findings comparing the different testing-approach patterns that the Japanese and U.S. vendor teams reported of their respective client teams with respect to team structure, thought processes, expectations, testing focus areas, and trust levels (Section 5.1.1). Because I did not have direct access to the Japanese client team and limited access to the U.S. client team, the findings presented here are solely based on the experiences shared by the vendor team members. In addition, I reflect on the reasons for those differences in Section 5.1.2.

5.1.1 Meaning of Testing: Different Approaches and Perceptions

As reported by the vendor teams, the Japanese Client (S3) Team and the U.S. Client (S3) Team teams differed in the approaches they adopted towards testing. Based on the experiences that the vendor teams shared, I identified four differences in their approach patterns, which I present in this section.

5.1.1.1 "Kensa" Testing Group

The Japanese Vendor (S3T3) Team mentioned that most of the Japanese client organizations would have their own quality-assurance group[1] known as a Kensa testing group. In Japanese language, *Kensa* means examination. The participants stated that this Kensa group did examine their testing capabilities. After the software was developed and tested by the vendor team and before it was shipped to the customers, the software would be delivered to the Kensa team for testing. The Kensa team was focused on assuring/verifying that the software met certain pre-determined and pre-decided quality standards. Thus, although I do not have data to verify whether Kensa testing is different from user-acceptance testing, I speculate that the two are different because the Kensa team performed thorough testing beyond performing the basic "sanity" testing, which involves just ensuring that the mutually agreed-upon requirements are met (which is usually the case with user-acceptance testing).

Every participant described the "nightmares" that occurred when the Kensa team detected defects[2] that they believed should have been discovered by the Japanese Vendor (S3T3) Team during the testing cycle. The "nightmares" occurred because it was a time-consuming task to regain the Kensa team's trust and confidence—to convince the Kensa team that, when a defect was uncovered, there were no other similar defects in the system.

[1]The words quality assurance and testing were used interchangeably by my participants as described by one of them "In the industry parlance when you look at the use of the word quality assurance, it is usually used all-encompassing, means including test execution. So the testing teams and quality assurance teams kind of go together. So it is used interchangeably."

[2]"Bug" is the term participants frequently used to refer to fault, failure, error, or defect. Hence, I use "bug" interchangeably to mean any of these terms.

The Kensa team's philosophy was that no one can test everything. Thus, they believed that when a defect is found in software, there is a high probability that similar defects would exist in the software that might be there because of some development or process weakness. To assure the Kensa team that other defects did not exist in the software, the Japanese Vendor (S3T3) Team had to undergo painstaking root-cause analysis procedures and create comprehensive documentation.

(Study S3)

The whole philosophy is that they [Kensa testers] cannot test everything. So they will just do some random testing. They find something, then they will start believing that there are other problems. Because there was some weakness in the person's development or the process because of which this is introduced. So same weakness would be somewhere else. So then you have to ensure them that this was the weakness and I identified this weakness and based on this I did a review of the application where similar kind of problems would have occurred and that kind of a thing. And I have to tell them that this is the final output. (Quote 15)

One such common root-cause analysis method adopted by the team was the why-why analysis method. Why-why analysis method is a structured mechanism of deep investigation of a problem by asking why at least to four or five levels. The rationale behind the adoption of the why-why method is that the Japanese members are primarily interested in getting to the root cause of the problem and they believed that why-whys method helps them identify and understand the root causes of the issues they encountered. Moreover, they believed that the problems usually caused because of some generic situations such as the team was under staffed, or the required expertise level was missing, or incorrect review procedures were followed and fixing these basic issues can help solve the bigger issues that get encountered.

Whereas the vendor teams reported the Japanese Client (S3) Team to have a dedicated quality-assurance team, similar teams were absent in the U.S. Client (S3) Team configurations. Also, the U.S. Client (S3) Team did not have its own quality assurance team. In fact, some participants mentioned (and a U.S. Client (S3) Team member confirmed) that the U.S. Vendor (S3T1) Team had introduced them to many software-testing techniques (e.g.,

automation testing) and their benefits, and demonstrated to them the importance of quality assurance. The U.S. Client (S3) Team participants also mentioned that the user acceptance testing was done by their business users in a less-structured and unplanned manner and that they used user acceptance testing as a mechanism to train their employees and not to critically evaluate the newly developed system. Moreover, unlike the Japanese Client (S3) Team, the U.S. Client (S3) Team team did not follow methods such as why-why analysis for investigating problems that occurred in the projects.

(S3T3.TM1)

[Researcher]: so if I understand it correctly, for the Japanese to come to a point where they get the sense of confidence from the vendors, they have to do the why-why process. But for Americans [before I complete the sentence, the participant starts speaking]
[Participant]: 1-2 levels of whys will suffice and they will be okay with that. They will mention go ahead and fix this [issue], resolve it... (Quote 16)
[Researcher]: so the process starts in almost the same way in terms of building up the confidence again but it's just that for the US it is sufficient up to 2 levels but for the Japanese because they are so structured the will go to those five levels and only then will they stop.
[Participant] - That is exactly what was that I also wanted to sayâĂęya.

One Japanese Vendor (S3T3) Team's technical-management level member, who had worked for both Japanese and U.S. clients, illustrated the difficulty in explaining the cause of a problem to the Japanese and U.S. clients:

(Study S3)

It's probably little easier to explain an American customer whereas something like a carelessness or negligence on part of somebody is very very hard to explain to Japanese. For them (Japanese client team) to understand it, it has to be very systematic. Their conceptualization is that people, even if they are doing mistakes, there should be a mechanism to check that, which is an expectation from everybody but they [the Japanese client team] are much more thorough. (Quote 17)

The discussion suggests that the Japanese Client (S3) Team was more thorough, than the U.S. Client (S3) Team, to building testing-related infrastructure and performing root-cause analysis to ensure high-quality software was being developed.

5.1.1.2 "Difference in the Level of Thought"

Based on the experiences shared by my participants, it was evident that the two client

teams had different thought patterns. The vendor team reported that the Japanese Client (S3) Team emphasized the negative scenarios in addition to the basic functionality scenarios: "in case of Japanese customers they will probably go to the success cases and normal conditions later after they have hammered [thoroughly tested] the software with conditions, which are less likely to prevail." Thus, the scenarios with unanticipated inputs, situations, and environments were emphasized over the expected scenarios for which the software was being built. However, the U.S. Client (S3) Team seemed to emphasize the basic functionality more than the negative (less expected) scenarios. One of the U.S. Vendor (S3T1) Team participant quoted his client's expectations: "I don't care about understanding the capability of [the vendor organization], that is whether they [vendor organization] can perform 100% testing [and] whether they are very productive [in testing activities] . . . all I care is to see whether the basic functionality in the system works or not."

One participant explained the difference between the kinds of scenarios that the Japanese testing would cover and that the U.S. testing would cover by illustrating it with an example:

> But the kind of thought process a Japanese [testing team] puts in and the ways they would think to break the product, they are quite stringent as compared to what Americans do. So they [Japanese testing team] would try out conditions which I haven't seen at least. . . American testers trying it out. They would think of scenarios like, which is one example that Iget to hear quite often, when everything is running they will just pull out all cords to see how the software application is behaving. . . there is a difference in the level of thought, when testing a software product specially, [that] Americans put in and Japanese put in. . . So [by] level of thought I meant that the boundary conditions . . . their focus is not only on seeing whether the product is working, their focus is always also on seeing if the environment, what they assume or what they expect, [is being handled properly by the product]. They would constantly try to build

those strategies.

The above discussion reveals the differences in the testing-related thought processes of the two client teams. The discussion illustrates that the U.S. Client (S3) Team emphasized covering the expected conditions (basic functionality) first whereas the Japanese Client (S3) Team emphasized covering the less expected (boundary) conditions first.

5.1.1.3 "Depth and Breadth of Testing" (Using Resources)

Because it is practically impossible to ensure that software is defect free, a practitioner concentrates on using limited resources (i.e., effort, time, and money) most effectively to uncover as many defects as possible. However, the findings provided evidence suggesting differences in the way these resources were being used by the Japanese Client (S3) Team and the U.S. Client (S3) Team. The vendor team's description illustrated that the U.S. Client (S3) Team used time as a constraint and tried to make the best use of the available time to perform the testing tasks. In contrast, the Japanese Vendor (S3T3) Team shared that Japanese Client (S3) Team used the testing tasks as the constraint, and they adjusted their working hours to ensure that they completed the testing tasks at hand. Participant S3T3.TM1 described:

> Like the way, [the] Americans work and the Japanese work, they treat it totally differently…the working hours and all that. The Americans would be like there is one week allotted for testing. This is what I can execute; I can execute 400 test cases. Then let's see which are the 400 test cases I want to execute. 200 probably are the success [test] cases and probably 200 are some destructive [test] cases, which will be touching the boundary conditions. But for the Japanese team with the same thing, let's say they have allotted one week's time…their approach, I imagine, will be that for my software to be certified [by Kensa testing team] these are the 1600 test cases that I have to execute and I have to execute them within this one week. So it's you know the depth and the breadth of the testing that must be there, they will try to adhere to that.

The Japanese Vendor (S3T3) Team mentioned that, although the expectation was not clearly discussed, the Japanese Client (S3) Team always expected the vendor teams to work

late hours to finish the tasks to which they had committed. The Japanese Client (S3) Team not only expected the vendor teams to work late but also they worked until late to finish a required task. The participants shared stories of their on-site visits to Japan where they observed the client teams working late.

Interestingly, the U.S. Client (S3) Team expected the U.S. Vendor (S3T1) Team to be more "realistic" in their goals indicating that they should not over commit to testing more than what is possible to cover in the allotted time. In fact, in one of the meetings, Iobserved that a client manager joked with the vendor team asking them when would they sleep if they committed themselves to testing so many test cases.

This discussion illustrates the difference between the expectations of the two client teams in the ways in which the significant[3] but limited resource of time should be managed to make testing more effective.

5.1.1.4 "Attention to Detail" (Making Assumptions)

(S3T3.TM1)
Attention to detail, that is one thing that one should learn from them [Japanese testing team]. You don't assume anything. Because most of the things that go wrong are based on certain assumptions... don't take anything for granted unless you know why it is happening. If there is a valid reason for something to happen then you know that this is going to happen. Just don't say that this is going to fall down so it is going to fall down. (Quote 18)

The Japanese Vendor (S3T3) Team reported that the Japanese client members were meticulous in every aspect of their work. The Japanese Client (S3) Team focused on every detail associated with the tasks: "How did you reach that point [conclusion], how many issues did you face, how did you solve it [issue], what will you do that you will not face those issues ... everything." Not only were they meticulous in focusing on the task details, but also they were systematic and scrupulous in investigating cause of issues. One participant from the Japanese Vendor (S3T3) Team mentioned that in a long term partnership with any clients, they would have build the confidence in the clients that if and when problems

[3]Rooksby found that time plays a significant role in the way testing is organized [113].

occur the vendor teams can resolve them. However, in his experience working with the Japanese clients, despite the long relationship, the Japanese clients would follow the same scrupulous investigation procedures when issues arised.

(S3T3.TM1)
With the situations arising with Japanese customers, whatever be the length of relationship [i.e., client-vendor partnership period] this [why-why analysis] is the process that they will follow. They will ask the five levels of why, they will discuss and debate, and only then they will buy the argument that the vendor has understood the problem. (Quote 19)

Interestingly, the U.S. Client (S3) Team appeared to be more focused on the end result. They were interested in understanding the processes, but they did not investigate issues in such detail as described by the Japanese Vendor (S3T3) Team. I observed that the U.S. Client (S3) Team was interested in understanding whether there were processes in place, but they were less interested in the learning about the process details.

Thus, my observations suggest that the Japanese Client (S3) Team was more process-focused than the U.S. Client (S3) Team, who appeared to be more results-focused.

5.1.1.5 Different Trust Levels

There is significant literature that discusses the role of trust in software development, particularly offshoring and outsourcing software development [21, 31]. I observed that there was an interesting interplay between trust and testing. The study data and observations implied that the Japanese Client (S3) Team did not trust others to perform their testing activities. The Japanese Vendor (S3T3) Team reported that it was always difficult to gain the trust of the Japanese clients. The Japanese Client (S3) Team would not trust others easily and it took effort for the Japanese Vendor (S3T3) Team to win the trust of their client team. The attention-to-detail (Section 5.1.1.3) and process-focused approaches of the Japanese Client (S3) Team (Section 5.1.1.4) support this point. Moreover, the previous discussions also suggest that the Japanese Client (S3) Team seemed not to trust that the users would use the software as expected or the environment would be conducive for the software to behave as expected. In contrast, the participants reported that it was relatively easy to gain the trust

of the U.S. Client (S3) Team. The U.S. Vendor (S3T1) Team reported that they did not have issues of trust with their U.S. Client (S3) Team. Additionally, I had access to the geographic distribution of clients who were using the participant organization's independent testing service. The data reported that 63% of the clients were from the U.S. but only 1% of the clients were from Japan. Participants also mentioned that the Japanese standards of quality were extremely high and thus, it was difficult to gain their trust in terms of testing and quality.

These observations imply that Japanese Client (S3) Team did not readily trust others for testing whereas the U.S. Client (S3) Team did not show reluctance in trusting others with testing activities.

5.1.2 Discussion: Reasons for the Differences

In Section 5.1.1, I reported the differences that I observed between the testing-approach patterns of the Japanese Client (S3) Team and the U.S. Client (S3) Team, as reported by their respective vendor teams, with respect to aspects such as testing-team structure, thought process, expectations, primary focus, and trust levels. Several questions arise from these observations, including: Why did the two teams approach testing differently? What are the reasons for these differences? Based on my analysis, I believe that testing is greatly influenced by culture and in this section I provide some evidence that supports and explains this point.

5.1.2.1 Uncertainty Avoiding Cultures Test Thoroughly

The activity of testing and the test results thereof (in the form of the number of defects) helps the software developer build trust in the software that is being built [17]. Testing is the means of gaining an assurance that the software will work, resulting in an increased level of confidence in the software. Testing can be viewed as a mechanism for avoiding ambiguity of how the software will behave in uncertain situations and thus, it helps in avoiding uncertainty. In Section 5.1.1, I discussed that the the vendor teams reported that

the Japanese Client (S3) Team significantly emphasized boundary-condition and negative-test scenarios. They paid ample attention to detail and did not make many assumptions, and they were reluctant to reduce their testing tasks despite time constraints. These observations indicate that the Japanese Client (S3) Team wanted to ensure that their software could handle as many uncertain situations as possible. They wanted to avoid any ambiguous situations by ensuring that their software was capable of handling those situations appropriately.

Uncertainty avoidance, described by Hofstede as a cultural dimension, is the "extent to which members of a culture feel threatened by ambiguous or unknown situations. [68, p. 191] Based on the tolerance levels to ambiguity, Hofstede assigned different Uncertainty Avoidance Index (UAI) scores to each nation. The UAI scores are interpreted in a certain way: a high UAI score indicates that the cultural group is anxious about ambiguous situations. Japan's UAI score is 92, indicating their higher level of anxiety over ambiguity and uncertainty. This high score may explain why the Japanese Client (S3) Team wanted to test their software so thoroughly: testing helped reduce their perception of risk of a software failure by avoiding ambiguous situations [68, p. 197]. They emphasized rigorous testing and quality standards to make uncertain events predictable and interpretable, thus, reducing their anxieties of failures.

In contrast, the U.S. UAI score is 46, which indicates a high tolerance for ambiguity and uncertainty. The "realistic" attitude of the U.S. Client (S3) Team towards testing may be due to their lower levels of anxiety that result from uncertainty. This lower anxiety levels may also be the reason for U.S. Client (S3) Team not monitoring the U.S. Vendor (S3T1) Team's activities as closely as the Japanese Client (S3) Team monitored the Japanese Vendor (S3T3) Team's activities—because they are relatively more comfortable with ambiguity. This cultural dimension seems to explain many of my study observations regarding both client teams' behaviors. The UAI scores hint at the reason why the Japanese Client

(S3) Team was so driven towards thorough testing—to avoid the ambiguous situations—and why the U.S. Client (S3) Team was not.

These findings may imply that cultural facets such as uncertainty avoidance dictate software-testing processes and practices (e.g., extent of thoroughness in testing) adopted by a team.

5.1.2.2 Quality is Based on Perception

The discussion in Section 5.1.2.1 suggests culture's role in the Japanese Client (S3) Team's commitment to thoroughness in testing for building high-quality softwares. Surprisingly, although the U.S. Client (S3) Team's testing activities appeared to be more relaxed (because they aimed to be more "realistic") than the Japanese Client (S3) Team's, they were satisfied with their software quality standards. Although the vendor organization was providing similar quality standards and services to both the clients, the Japanese Client (S3) Team appeared to have a higher software-quality expectation from their vendor team than the U.S. Client (S3) Team. This difference indicates that the client teams' quality-related expectations differed although both clients wanted good-quality software.

Beizer states that when a test passes, it does not change the software's quality, but it changes our perception of its quality: "If a test is passed, then the product's quality does not change, but our perception of that quality does [17, pg.6]. Beizer's statement implies two ideas. First, quality is subjective and it is based on individuals' perceptions. Second, more passing test cases can positively change the perception of the software's quality. One participant explained that "...quality is very much based on your perception. Something that flies [works] well with person A, [that is] he might be very satisfied and very happy with the product; person B might not really be [satisfied and happy]. He might have certain different expectations, and he might rate that as a low grade quality product." This quote indicates that quality is viewed subjectively in the industrial setting, and it provides insights about the reason the two client teams had different expectations with respect to quality.

Interestingly, much of psychological and anthropological research has shown that aspects, such as perception, values, and perspectives, are shaped by culture [124, 50]. A popular example includes the perception of color, where white used in many Western countries is seen as a symbol of innocence and purity. The same color, in many Eastern countries, is seen as a symbol of mourning. The above discussion indicates that a cultural facets influence the perception of quality [17].

This discussion raises the question that when different cultural groups evaluate the same product, will they certify the product quality differently. Further research is required to explore and investigate this thread.

5.1.2.3 Bug's Discovery and Severity is Culturally Influenced

In the Sections 5.1.2.1 and 5.1.2.2, I illustrated how culture is influencing the testing practice in subtle but significant ways. In this section, I demonstrate another example of the influence culture can have on the testing practice.

Beizer states that a defect's severity depends on the context and the culture in various ways [17]. First, he illustrates that a defect's severity depends on the *context or situation*: for the same bug (a round-off error in an orbit calculation) with the same symptoms, the severity will depend on whether the bug appears in a space-ship video game or a real space shuttle. Second, he exemplifies that a defect's severity depends on the *software developer's cultural aspirations*: a test tool vendor is more sensitive to bugs in his product than a games software vendor. Third, he explains that *user's cultural aspirations* also determine a defect's severity: a research and development team would accept a defect with a work around but a banker can be sued for the same defect.

I speculate that not only is the defect severity culturally dictated, but also the defect discovery is culturally determined. It is widely known that it is impossible to build a software that is defect free. Most software-testing effort can be viewed as an effort to expose the defects that are likely to occur. Exposure of a bug and its likelihood depends on the

way software is used in a particular situation or environment. Although, there may be no control over the environment, there is definitely the users' assumptions and usage patterns that determine whether a particular bug will be uncovered. For example, consider a web application that accepts a postal address as input. If this web application were being used in the U.S., then there is a low probability that a user will add a landmark such as "opposite Standard Chartered Bank" in the address because the U.S. postal addresses have a standard format. However, if the same application were to be used in a country such as India, where they lack such standard formats, there is a high probability that the address window of the application will receive such landmark entries.

Thus, it is not important to test for the various combinations of valid landmark entries for a U.S. user group. However, it is crucial to test for those entries for an Indian user group. This example also illustrates that if there were an address parsing defect associated with the landmark-related information, it is likely that an Indian tester (end-user) would encounter it rather than a U.S. tester (end-user). This difference occurs because both testers (end-users) have different cultural exposures, and thus, their assumptions and usage patterns differ. Eaton explains that the "domain knowledge or detailed business knowledge is deeply tacit, and represents the local familiarity an IT [Information Technology] worker has with the context or culture for which a technology solution is developed" [49]. The U.S. tester, not familiar with such postal address formats, might not think of writing test cases for such input data. This unfamiliarity might result in a defect not being found during testing and being discovered after deployment by an Indian user. In contrast, the Indian tester will probably write test cases for such postal address formats but those situations are unlikely to occur for a U.S. user, resulting in useless effort. Hind et. al., argue that "adoption and appropriation practices are likely to lead to technology (e.g., software) use that is particular to the cultural context" [65]. Consequently, because technology (which includes software) usage patterns are greatly culturally influenced [65] (which forms the basis of user-centered design), technology testing patterns (which includes defect's discovery) may

also be culturally dictated. In other words, when the same application is being tested by different cultural groups, there is likely to be a difference in the kind and number of bugs uncovered by different groups because their adoption and usage practices will differ culturally. This idea suggests that testing is cultural but future research is required to investigate the extent of culture's influence.

5.1.2.4 Culturally Driven Testing Thought Process

"Testing is a process in which I create mental models of the environment, the program, human nature, and the test themselves. Each model is used until either I accept the behavior as correct or the model is no longer sufficient for the purpose. [17, p. 22]." This idea implies that testers first start with some mental model of the way software will be used in a particular environment. Consider the following example that one participant narrated regarding a file. There were several scenarios that the Japanese Client (S3) Team expected the users would encounter:

> I had to read some data from a file. The data in the file is arranged in the key-value pair. There is a key to look for and then I look for the value itself corresponding to that key. So what Kensa [Japanese client's testing team] tested for was first the file itself is not there, then that key is not there, then the key is not there in the first line... let's say if the key is uppercase, lowercase ... if the key has some different [number of] letters... [if] the value provided is key=nospace=value then how will the product function. If the file is read-only, if the folder is read-only ... they tested for all these situations.

This narrative highlights the likely mental models that vendor teams believe that the Kensa team seems to possess. The Japanese Client (S3) Team probably tested for these scenarios because, in the past, they had encountered these or similar situations. Thus, their mental models of testing have expanded to expect such situations to occur. However, the participant reported that they had not thought of all these scenarios. I speculate the

reasons for their lack of thinking are deeply rooted in the models that the testing teams possess. Beizer states that the "art of testing consists of creating, selecting, exploring, and revising models. Our ability to go through this process depends on the number of different models I have at hand and their ability to express a program's behavior." However, Shore argues that these mental models are influenced by culture to form what he calls the cultural (conventional) mental models that are shared with a group of people. The participant and his team may not have been able to think of those scenarios as thoroughly as the Japanese client team did because they might not have encountered such usage patterns. Hence, they would not have expanded their testing mental models.

This implies that different cultural groups may possess different cultural (conventional) mental models of testing (i.e., the environment, program, human nature, and tests), which help them think about testing scenarios differently. Although I do not have sufficient data to make any strong conclusions, I observed that for the similar (SAP) software, the Mexican Client (S3) Teamand U.S. Client (S3) Team did approach the testing differently. Additional studies are required to explore this thread further.

5.2 Narrating Client-Vendor Cultural Clash Experience

The previous section discussed the client-side cultural practices with respect to the realization of the software-testing activities and highlighted the manner in which the cultural facets influence the enactment of the GST practice. In this section, I present a vignette from the GST practice I studied at the Vendor V3 organization. The vignette is associated with an escalation situation that occurred at this organization a few days before I started Study S3. I describe the vignette in detail and discuss the reasons for escalation as illustrated by my participants. This episode from the practice highlights many underlying customary and habitual patterns that are embedded in the GST practice. A detailed description of the episode from the perspective of the participants will facilitate a rich discussion of the cultural meanings embodied in the practice, which I present in the next chapter (i.e.,

Chapter 6).

5.2.1 Describing the Escalation Situation

In GST project that I studied, there were typically three or four testing cycles. Each cycle was four weeks long. Before the testing cycles began, the client and vendor teams agree upon a testing plan for test cases execution. In each cycle, pre-determined number of test cases are targeted for testing and the vendor teams are expected to finish that amount of testing. The test cases are distributed over the four weeks duration and at the end of each week, the vendor team is expected to send the testing progress status to the client team.

When I started the study at the Vendor V3, the Mexican Vendor (S3T2) Team had just finished its first cycle (i.e., Cycle one) of testing. The project, at that time, was in critical state, because I was told that a few days ago, a big escalation with higher management had occurred. I took this opportunity to investigate the escalation situation, which I describe here along with a discussion of the causes of the escalations from the participants' perspectives. I investigated the situation from the perspectives of all the three levels depicted in the Figure 5—the delivery, project, and technical management level perspective. An interesting pattern of perspectives emerged that indicated the visibility of the members at different levels.

The Mexican Client (S3) Teamhad asked the Mexican Vendor (S3T2) Team to perform some testing in the Cycle one, which was to be finished in four weeks. In the test planning phase, there were test cases allocated for each of the four weeks and the vendor team were expected to share the status of the testing progress with the Mexican Client (S3) Teamat the end of each week. However, at the end of week one, the testing status showed many red flags, indicating that much of the testing that was planned to be finished in the week one was not done as scheduled. This concerned in the client team, and they started investigating the matter by performing some audits. In the audits, other problems emerged, which blew the situation out of control and the issues were escalated to higher management.

5.2.1.1 Client Side of the Story

Parent (U.S.-Mexico) Organization use to follow the practice that the Parent (U.S.-Mexico) Organization's team members must also perform the first round of testing. There was one business reason for adopting such a practice: the senior managers at the parent organization wanted the client-side team members to be accountable and responsible for the testing that was being planned and executed at the offshore, vendor organization. Their reasoning was that if the first cycle of testing for a new functionality is done by the Parent (U.S.-Mexico) Organization's team, they will be better informed of the requirement specification document's contents:

(S3.CO1)

. . . the first time new functionality is to be tested, [client manager] wants the [client] team to test it not the [vendor] team. And what that really is, it's a check off that what they [client team] puts in their specs [requirement specification document] is really what they are seeing and what they are doing (Quote 20)

.

Additionally, the Mexican Client (S3) Teamwas using testing as a means to train their employees (as described earlier in Section 4.1.1). So, they were simultaneously performing testing in the "training" environment, while the Mexican Vendor (S3T2) Team was performing testing in the "official" environment. However, consequently, the Mexican Client (S3) Teamraised concerns about Mexican Vendor (S3T2) Team's productivity and expertise level because they had finished the testing tasks before the Mexican Vendor (S3T2) Team was able to finish. The productivity was a concern because the Mexican Client (S3) Teamwere expecting the vendor team to finish sooner given that the Mexican Vendor (S3T2) Team claimed to be experts at testing the system and technology that the client team were working on. The expertise level concerns were raised because the Mexican Client (S3) Teamwere concerned when they were asked clarification questions regarding the execution order of the steps during test-case execution. These questions indicated that the Mexican Vendor (S3T2) Team were not as experts as they were claiming.

5.2.1.2 Vendor Side of the Story

The Mexican Vendor (S3T2) Team's side of the story was that they were indeed happy that the escalation happened because they also had several concerns with the Mexican Client (S3) Teamafter the cycle one of testing started. First, the Mexican Vendor (S3T2) Team had difficulties in acquiring the essential information and required infrastructure to perform the testing on time, which caused delay in the testing schedule. Because of the lack of these resources, they lost four out of five days in the first week of cycle one of testing, which led to the red-flag indications in the weekly status report. Second, Mexican Vendor (S3T2) Team mentioned that the test planning itself had some flaws, which they had tried to highlight to the clients during the test planning phase. However, the Mexican Client (S3) Teamwas adamant about the plan and refused to change it. Lastly, Mexican Vendor (S3T2) Team admitted that the resource planning from their side included many junior (e.g., novice test engineers who have less experience working on a given technology) test engineers on the team, which resulted in the drop of their expertise level, causing additional delays in the test executions.

5.2.2 Discussion: Reasons for Escalation

I investigated the escalation issue to understand the participants perspectives on the reasons for escalation. In this section, I discuss the reasons that the participants shared which were the contributors to the escalation situation. Participants mentioned that several factors were involved for the cumulative failure of the cycle one. Two senior managers—one at delivery management level (S3.DM2) and other at project management level (S3T2.PM2)—coherently summarized the reasons for failure during two separate one-on-one interviews I conducted with them. The conversation with the S3.DM2 is as mentioned below.
(S3.DM2)

And that's where the communication gap was very problematic. They said yes and we said okay [emphasized "okay"] rather than saying that [its] not okay. And this is what exactly happened. By the end of the first week we were no where near the 60% [target that was committed] time. And that's what the escalation was all about—[Vendor Organization] does not have the capability, they do not have the communication skills, they do not have the attitude . . . The thing is that globally not everybody understands offshoring [global delivery model]. Second thing is that if [Vendor Organization] thinks that they have the capability and they have the experience, they need to use their powers to effectively communicate their problems rather than randomly going hand-in-hand and agreeing to what clients are expecting because that is going to lead us into trouble. Local hires not adapting to global delivery model is a problem, local language is a problem. (Quote 21)

5.2.2.1 "First Week Lets Load Less"

One reason of failure the vendor team reported was the clashing planning strategies. The vendor team wanted to plan testing such that there was less work assigned for the first week:

(S3T2.PM2)
Ok, for (test) planning . . . we always plan 10-15% in the first week, 20% in the second week, slowly gradually increase, so by that time the environment stabilize the data issue is resolved, so we get the speed. It should be a curve; it cannot be a flat projection. We tried to do that curve projection but the customer pushed for this flat projection. They said 'no, we need everything' and we are not able to convince them. We were not able to pushback on them saying that this is not the right way of doing it and that is where the problem was. (Quote 22)

Based on the participants' description, this was accepted and preferred practice— having less testing activities in the scope for the initial weeks and having more activities in the later weeks when the environment is stable and infrastructure is available. The vendor-side managers insisted on having such a planning strategy, particularly for new clients, because based on their past experiences they have encountered several testing-setting related issues (e.g., absence of infrastructure related information) in the initial weeks, which significantly caused delays in the testing activities.

However, in this case, the Mexican Client (S3) Teamseemed to be adamant on having more testing planned than the amount recommended by the Mexican Vendor (S3T2) Team for the initial weeks. Moreover, the Mexican Vendor (S3T2) Team were "not able to convince them" when the client team insisted for more test cases in the first two weeks. Consequently, they decided on planning 60-65% of the test case execution in the first two weeks and the remaining in the last two weeks. One participant stated that

(S3T2.PM1)

"We wanted to have this communication through to the [client team] that first week let's load less and let's pick up next week onwards. [However,] those guys [client team members] were adamant that pick up as much as possible in the first week and then slowly slowly reduce it. So they picked up [assigned] almost 60% of the targeted testing in the first week." (Quote 23)

In addition to the clashing planning strategies, the vendor team members mentioned that they implicitly assumed that the latter two weeks would be buffer weeks, which they planned to use to finish the leftover testing that they were certain they would not finish as scheduled in the first two weeks. However, the client team had a different assumption with respect to the last two weeks—they assumed that this was a buffer week and would most likely not be used because the testing will proceed as scheduled.

Thus, the two teams had assumed and interpreted the testing plans differently, which caused a mismatch in their understanding of the expected outcomes. This difference was one factor that led to the escalation.

5.2.2.2 "Our Team Was New"

The learning model at the vendor organization consisted of (1) classroom training and (2) on-job training. The class-room training, which was typically rendered to fresh college graduates, was considerably beneficial to the non-computer science majors because it focused on fundamental computer-science related concepts. Nonetheless, on-job training was crucial for all newly joined engineers because this was the training that provided the coaching required to make meaningful contributions in a project. I observed that at the vendor organization V3, the test engineers, new to the technology, learn the technology

mainly on the job while working with other team members who are experts. Although this was a common practice that the vendor-organization teams follow, they do not necessarily mention this practice to their client teams because they do not want to lose the client's confidence in their expertise. Nonetheless, the vendor teams are confident that they can perform the tasks with the novice test engineers on the team because their strategy is to plan the testing activities accordingly, taking into account the team members' expertise. However, in this particular escalation situation, the planning did not occur as per the Mexican Vendor (S3T2) Team's expectations (as discussed in Section 5.2.2.1). This difference in the actual and expected testing plan became a recipe for lack-of-expertise exposure and a consequent failure.

THe events occurred in a manner that exposed the lack of expertise of some team members to the client, leading to shattered client's confidence in the vendor team's capabilities. For instance, the project lead, a domain expert, was on his vacation during this time. Consequently, a novice member of the Mexican Vendor (S3T2) Team asked some basic clarification questions directly to the Mexican Client (S3) Team, which revealed his lack of expertise level. This information was considered as basic domain knowledge and hence it triggered concerns among the Mexican Client (S3) Team. In an ordinary situation, the team members would have waited to ask these questions to their expert on the team, who would have bridged this knowledge gap. However, the absence of the expert exposed the other members' lack of expertise leading to raised concern levels: "The biggest mistake was to put non-experienced people in the team and said [say] to the client that they were experts. Client is not dumb and they realized that [vendor organization] offshore team did not have the experience promised" (S3.OC1).

5.2.2.3 "Local hires not adapting to global delivery model"

In the case of the Mexico client-vendor project, the onsite coordinators were the "local hires"—they were Mexican local recruits working for the vendor organization. These onsite coordinators were new to the GSE setting and they were unaware of the "global delivery model"—which encampasses the customary practice adopted by the GSE practitioners at the Indian vendor organizations (I discuss this model in detail in Section 6.4). The offshore vendor team also was not consciously aware of the hidden dimensions that governed the this practice. Hence, they never considered it important to explain the Mexican (S3T2) Onsite Coordinator Team the internalized meaning of the GSE practice. However, the escalation situation exposed Mexican (S3T2) Onsite Coordinator Team s' lack of knowledge and understanding of the meaning of the GSE practice. The offshore vendor team realized then that the onsite team did not understand as well as follow the cultural norms associated with the model.

For instance, frequently, Mexican Vendor (S3T2) Team had to be flexible in terms of work hours and the duration. For the offshore vendor team members, working extended hours and providing flexibility in work hours was a common and acceptable GSD practice. However, for the new Mexican onsite coordinators, there was much reluctance and resistance to work at odd hours and for extended durations. When questioned why they were hesitant, the senior managers explained that they were "governed by the law of the land." I found that this inflexibility of Mexican (S3T2) Onsite Coordinator Team led to issues in "handshaking," which refers to the procedure of systematically transferring the knowledge from members located at one geographic locations to members located at another geographic location. Handshaking is important because it helps ensure uninterrupted functioning of the follow-the-sun practice, which is crucial factor for GSD's success. The hesitance to work at flexible hours made it challenging for the offshore team to get the required information from onsite coordinators resulting in they being unclear of the expectations of the client teams, which consequently caused struggled in understanding the tasks

that they were responsible for executing at the offshore. This ambiguity led to confusion, which caused further chaos and delays resulting in an additional reason for escalation.

5.2.2.4 "We were not able to pushback"

My study insights revealed that there were clashing planning strategies being proposed by the client and vendor teams. One the one hand, the Mexican Client (S3) Teamwas insisting on planning the testing activities such that they wanted more testing to be done in the initial weeks than later. On the other hand, the Mexican Vendor (S3T2) Team wanted to plan for less testing in the initial weeks.

The Mexican Vendor (S3T2) Team were subject-matter experts in the technical domain of client's project and they had significantly more experience working in the GST setting than their clients. Ironically, the Mexican Vendor (S3T2) Team stil agreed to adopt the planning strategy preferred by the client, despite their reluctance. and they having the experience working with the GSD model and practice. This reluctant submission to the decision of the clients was expressed in the participant's statement:

(S3T2.PM2)
Leads were not able to convince the customer [client] that this was not the right way of doing, so that was one of the reason that we over committed in the first two weeks and couldn't deliver. Definitely, its not right. That was definitely the big issue. (Quote 24)

The statement indicates the Mexican Vendor (S3T2) Team's failure to convince the Mexican Client (S3) Teamabout the shortcomings of their planning strategy, which was bound to fail the testing plan. This scenario depicts how the vendor team was convinced about the certainty of failure of the client's planning strategy, but despite knowing this information, they did not "pushback" and persuade the clients to change their strategy.

5.2.3 Post Escalation Event: Consequences

5.2.3.1 "[Client] team is saying no."

Several weeks after the escalations happened, I conducted an observation of a weekly status-update meeting, where the Mexican Vendor (S3T2) Team was updating the project's

progress to the U.S. client manager (S3.CO1). The client manager was particularly interested in the progress because she was representing the testing initiatives at the Parent (U.S.-Mexico) Organization. Hence, she was inquisitive to know the progress of the Mexican testing project mainly because it was a relatively new client-vendor engagement and, based on this project's success, new testing initiatives were expected to be started with the Vendor V3 organization.

The practice of client team testing first time, discussed in the previous section, provided a sense of security to the Mexican Client (S3) Team; this security was required because they were not quite confident about the Mexican Vendor (S3T2) Team's capabilities given that they had newly engaged in this GST project business model. The comfort came from the fact that they still had the opportunity to perform the testing of new functionality first, which is usually the more critical module with most of the defects in the initial rounds of testing. They could now consider the Mexican Vendor (S3T2) Team as a support team that helped them performing testing on the existing functionality, which is comparatively more stable.

However, during the weekly update meeting, the team had started discussing one issue that had been concerning the client manager (S3.CO1). The issue was that the Mexican Client (S3) Teamhad continued to perform testing activities themselves in the subsequent cycles of testing rather than allowing the offshore vendor team to perform the testing. They were now testing not only the new functionality, but also the other functionalities. These actions had concerned and confused the vendor team as well as the client manager (S3.CO1): "looking at the [X Module] I wasn't expecting the [client] team, in cycle three, to be running so many test cases." A few weeks before this meeting, the client manager had visited India, and she was informed that "cycle three is when the [vendor team] will be testing everything." However, during this meeting, she was being informed that the Mexican Client (S3) Teamwanted to do considerable amount of testing. Later in the meeting, the different members attending the meeting—offshore vendor team, client manager, and the onsite

coordinators—contemplated the reasons behind the Mexican Client (S3) Team's thought process and concluded the meeting after deciding that the onsite coordinator will try to convince the Mexican Client (S3) Teamto send more testing activities offshore.

Immediately after the meeting, I met with one project-management level manager—S3T2.PM1—who had attended the meeting. I followed up on the meeting's discussion to better understand the reasons behind the Mexican Client (S3) Team's decision. He explained the situation to me as

(S3T2.PM1)
[Client's] high-level business kind of feels more confident if their team does the testing. It is just the kind of comfort factor because [Vendor V3] is finally any third party vendor for them... In fact this is the first time in Mexico that [Vendor V3] is doing the testing. So far entire testing used to be done only by the [client] team in Mexico. So this is the first time they said that okay [Vendor V3] can also do the testing but,... because they wanted to have some kind of more confidence, they said that 'in any cycle any new functionality we will test. You only test the existing functionality.' So that's why in cycle one, they tested every new functionality and we tested existing functionality... Now S3.CO1 was expecting that in cycle three, all the new functionality should have already been tested by [the client team], and so in cycle three [the vendor team] will do all the testing. But this [client] team is saying no. (Quote 25)

The above conversation (25), illustrates the reluctance of the Mexican Client (S3) Teamto believe the Mexican Vendor (S3T2) Team's capabilities to perform the testing: "feels more confident if their team does the testing." S3T2.PM1 indicates that it was the first time that Mexican Client (S3) Teamhad offshored the testing activities to the Mexican Vendor (S3T2) Team. The reluctance to believe in a new partnership is not a surprising behavior. However, what is thought provoking here is that the Mexican Client (S3) Teamhad designed a plan to handle the reluctances associated with the first time offshoring experience: they had decided that in any cycle the new functionality would be tested by the Mexican Client (S3) Teamteam first. Following this plan, in cycle one, the Mexican Vendor (S3T2) Team team tested the existing functionality.

However, cycle one was the cycle when the escalation happened. This indicates that cycle one was in fact a critical time for the Mexican Vendor (S3T2) Team team to invest

effort in building a good rapport and relationship with the clients. Instead, in cycle one, the team decisions and actions (e.g., putting novice engineers on the team and portraying them to be experts to the clients) demonstrated ignorance to being sensitive about the credibility aspect. This ignorance made the situation worst and they were set on the path that was bound to fail. Thus, the key issue here was the Mexican Vendor (S3T2) Team's negligence towards handling their credibility cautiously. Consequently, the client team lost trust in the Mexican Vendor (S3T2) Team, leading to the client team wanting to do most of the testing even in cycle three.

5.2.3.2 "But we have delivered"

After the escalations, the Mexican Vendor (S3T2) Team worked overtime and delivered the activities on schedule as committed by them. The Mexican Vendor (S3T2) Team's senior managers (project management level) were hesitant to accept that their actions had jeopardized their credibility in front of the client. They justified their stance by explaining that "Some dips [mistakes] may have happened, you have to accept the fact. But we have delivered, so that is [has] not resulted in... we losing the [client's] credibility.(S3T2.PM2)"

There is evidence in the study data that suggest that Mexican Vendor (S3T2) Team tacitly assumed that as far as they completed the testing and delivered the test results to the clients within the committed deadline period, there were no repercussions of any failed situations. Moreover, they had strongly internalized this idea that overtime working, which appears to demonstrate that they are committed to their work, would overshadow any credibility problems: "we are not losing credibility we are somehow surviving, surviving, surviving because our team is ready to put that extra effort, we are really committed. The consequence is that we may burn out eventually. That is a right consequence but losing a credibility from customer, this does not happens (S3T2.PM2)."

Ironically, I found that the vendor participants perceived the Mexican Client (S3) Team to be rigid and less trusting of the Mexican Vendor (S3T2) Team's capabilities. In other

words, the Mexican Vendor (S3T2) Team believed that gaining confidence of the client was a big challenge. This challenge was one of the reasons why the Mexican Vendor (S3T2) Team had recruited local Mexican engineers, instead of the typical practice of deploying offshore (Indian) team members onsite, to work closely with the client team as onsite coordinators.

5.2.4 Summary

Thus, this narrative presented the perspectives, thoughts, and beliefs of the Mexican Vendor (S3T2) Team relating to what happened before, during, and after the escalations. However, this discussion still does not explain the reasons behind these thoughts and behaviors, and beliefs. To understand these reasons, it is important to investigate the underlying cultural systems that are embedded in the GST practice and govern the enactments that realize the GST practice.

CHAPTER VI

CULTURAL MODELS AND THEIR INTERPLAY

Holland and Quinn define cultural models as the "presupposed, taken-for-granted models of the world that are widely shared... by the members of a society and that play an enormous role in their understanding of that world and their behavior in it" [69, pg. 4]. From this perspective, cultural models are constituent attributes of both culture and cognition, characterizing a "twice-born nature" realized in the world as well as in the mind [124]. Such models comprise the realms of culturally-constructed knowledge and they facilitate implicit communication and meaning-making of experiences within the society where the models reside [69]. In the context of my research, this society is the society where GST is practiced and GST practitioners (e.g., my participants) are the members of this society.

In this chapter, I highlight some cultural models that were embodied in the GST practice and that were internalized by my participants. The purpose of this chapter is twofold. The first purpose is to explain why my GST-practitioner participants held certain sets of beliefs and thoughts, which governed their behaviors. I explain the reasons with an illustration of the four different cultural models that emerged from the three studies (mainly from the third study) (Sections 6.1, 6.2, 6.3, and 6.4). The second purpose is to demonstrate how my participants' beliefs and ideals function in the practice. To demonstrate this functioning, I use these cultural models and unfolded the escalation situation, described in the previous Section (Section 5.2), and illustrated how these cultural models interplay in the GST practice. (Section 6.5).

Because I had the opportunity to do extensive studies only at the vendor side, I was able to capture only the vendor-side cultural models. Hence, this chapter describes primarily the vendor-side cultural models uncovered through the study. Additional studies are needed to

identify the cultural models that are embedded in the client-side practice.

6.1 Agreement Model

The Agreement Cultural Model refers to the degree to which one culture/group agrees with others, sometimes despite an underlying reluctance. This means that one group may consent to the other group's decisions, ideas, and views, despite sometimes having a different point of view or thought about the same topic.

The Agreement Cultural Model is most commonly referred to as "always say yes" in much of the literature on culture and global software engineering [31, 152]. For example, Casey reports on the "always say yes" behavior of some cultures [31]. For another example, Winkler and colleagues discuss the rigid hierarchies in Indian organizations that result from the power distance in the Indian culture as an explanation of the "always say yes" behavior [152]. The Agreement Cultural Model is similar to the degree of activity or passivity, which refers to the "extent to which individuals in a culture see themselves as doers (active shapers of the world) or bearers (passive reactors to the world)" [89].

Participants from all the three vendor organizations I studied appeared to have internalized the Agreement cultural model because I found my participants demonstrating similar behaviors across the organizations. For instance, as described in the narrative from Study S3 (Section 5.2), the Mexican Vendor (S3T2) Team had agreed to the testing plan that the Mexican Client (S3) Teamwas insisting on, despite their reluctance. Later, they even admitted that "they were not able to pushback." This sentence indicates that the Indian vendor team wanted to disagree, yet they failed to disagree with what the client team was proposing. Thus, it appeared that the vendor team had failed to convince the client team to change their testing plan.

Additionally, I found a similar enactment of the cultural model in Study S2. In this study, the U.S. Client (S2) Team wanted the U.S. Vendor (S2T3) Team to test a particular module. They requested that the vendor team execute 100 test cases to test the module.

Although the U.S. Vendor (S2T3) Team members believed that they had better ways of performing this testing, the idea was not conveyed to the clients. One participant mentioned that he could effectively test the module using only five or six test cases rather than wasting time by executing the 100 test cases that the clients requested. The participant mentioned that:

(S2T3.TM1)
So for 5-6 [test] cases, I am going to write 100 some test cases . . . this number will look good but you know it's not logical work. I mean to say it won't be smarter work. Instead of having 100 test cases why don't you have 5-6 test cases. . . [However,] the goal will be the same but effort and visibility will be more in 100 test cases. (Quote 26)

This vignette depicts the classic behavior of the vendor team members—instead of proposing this improvement suggestion to the clients, the vendor team member adhered to the client team's request and avoided the possibility of correcting the clients [31, 118]. Thus, again, such a behavior illustrated the internalization of the Agreement cultural model that appeared to have been governing the participant's behavior.

Experience of Unproductive productivity. Interestingly, this internalization of Agreement Cultural Model has revealed a different set of experiences that I refer to as *unproductive productivity*: experiences in which the test engineers are "productive" in the sense that they produce the results that the clients requested, despite their belief that they are being unproductive in producing those results [120]. Identifying and avoiding such instances is important because it will help (1) increase productivity and reduce boredom for the vendor teams, and (2) increase cost effectiveness for the client teams by avoiding the elements that generate the unproductive experiences.

The reason for this unproductive-productivity experience can be explained by investigating the culturally-influenced behavior and thoughts of participant S2T3.TM1. The goal in this example was to test a particular module to uncover hidden defects. From what the vendor-team member mentioned in their interviews, the client teams translated this goal into numbers related to testing activities (refer to Section 4.1.5). Consequently, the

meaning of productivity for them was the number of test cases created/tested over a given time period. Interestingly, the vendor team appeared to have adapted to the client's interpretation of the goal. This team focused predominantly on completing the tasks given by the client team, accommodating their vendor team's meaning of productivity, namely the extent to which tasks requested by the client teams are accomplished. Choosing this meaning of productivity was a manifestation of the internalized Agreement cultural model. The difference in productivity perceptions led to this unproductive-productivity experience, particularly because this difference was not communicated and negotiated between the two teams. I speculate that there was a lack of communication because the teams have not been consciously aware that such differences existed. This is the point where the cultural system is said to have subsumed the technical system of practice because the cultural behavior governed the technical thought of improvement.

Such a subsumption was distinct in the later part of the conversation with participant S2T3.TM1. I asked the participant whether he tried to convey this issue to his senior managers or the clients. He replied that he had tried discussing this with his manager, but also added that:

(S2T3.TM1)
Why should I talk? Because you know, [the on-site coordinator] has provided everything to the client. If I go and reduce the estimated hours to 20 [which translates to the reduction in the number of test cases], everything goes wrong for the [on-site] coordinator, [project] lead, and they will be bouncing back on me only [laughs]...they can have so many reasons to defend [themselves]. (Quote 27)

The test engineer's choice of words indicate that, although he believed that there were better ways of performing the testing activity, he was reluctant to exert extra effort to make things efficient because he sensed that there would be some resistance from his colleagues. This conversation also reflects two attitudes: (1) as far as he is concerned, he has to complete the assigned tasks so that he shows he has been productive and (2) he wanted to avoid confrontations to maintain the team's harmony.

To explain this example better, consider a hypothetical situation where the clients still

perceive productivity as the number of test cases tested. However, in this hypothetical case, the vendor team did not perceive productivity as the extent to which tasks were completed because they had not internalized the Agreement cultural model. Instead, their meaning of productivity was related to the how many bugs were uncovered because they had internalized uncertainty avoidance related cultural values similar to the ones adopted by the Japanese teams (refer to Section 5.1.2.1) [118]. In other words, instead of interpreting the goal of testing a module as completing the execution of 100 test cases, they translate it to uncovering some number of bugs in the module. In this case, the vendor team might find that the test cases given by the client team are not sufficient and they might execute additional test cases, or that their test cases are redundant and hence they might not execute all 100 test cases. In either situation, this hypothetical vendor team will probably not experience the unproductive-productivity situation described above. Thus, "one of the reasons that solutions to similar problems evolve differently in different national cultures is because the assignment of meaning varies and therefore how the problem is framed and the practices that are appropriate vary" [65, p.159].

Hesitant to Always Say Yes. The above example of unproductive-productivity also appears to expose an alternative, conflicting model that is opposing the always-say-yes model. The clashing behavior generated from the conflicting model is highlighted in the sentence: "but you know it's not logical work. I mean to say it won't be smarter work" (refer to Quote 26). Other participants also expressed similar hesitance towards this cultural practice of always saying yes. One participant discussed an incident in which she was expected to meet an infeasible deadline only because the clients were promised it earlier. She clearly expresses her disappointment
(S2T3.TM2)

Whether the clients are like that or the managers, we don't know. It's also like we don't know where the things are getting masked . . . it's not like we are not reaching [meeting] the deadlines or we lack the information, skills, or we are not punctual. It's like we have practical concerns and we are not understanding why the client is not understanding these practical things. (Quote 28)

Her repeated questioning indicates her urge to explain the infeasibility to the clients (which means saying no to the clients or negotiating with them for doing a certain tasks), and shows a glimpse of the reluctance to always agreeing.

Although I was unable to capture these alternative conflicting models in detail (because of time constraints), the clashing behaviors hint that the vendor's cultural model is gradually transforming such that the always-say-yes practice is now being questioned. Such examples demonstrate that the culture (and so the cultural models) may be thought of as a process that transforms over time, rather than static dimensions (such as Hofstede's), which illustrate the evolving nature of culture. The question remains, however, of why the agreement (cultural) models continue to dominate the behaviors of the participants, while the other conflicting models have started emerging, despite participants' experiences such as the unproductive-productivity situations.

Mitigating Business Aspirations. Interestingly, the Vendor V3 organization appeared to have been aware of the existence of this internalized Agreement cultural model and hence it was making a conscious attempt to change this cultural behavior. One distinct illustration of this effort was the organization's mission statement, which included the following sentence. **(Organization Document1)**

For the success of our client organizations we may need to offer advice that is contradictory to what the client wants to hear. . . Bottom line is that the client is not always right. . . We should challenge ourselves to always have an independent opinion on the topic we are discussing with our clients. (Quote 29)

The statement—"we may need to offer advice that is contradictory to what the client wants to hear"— indicates that the organization has been making an effort to break out from the Agreement cultural model's influence by taking appropriate measures to create awareness of the model's embodiment in the everyday GST practice. Interestingly, this

organization was in the process of changing their business position in the market from being a service provider to becoming a business consultant to the its clients. I suspect that this change is the reason for the organization's attempt at reforming the cultural practice in the organization. Mention of statements, such as the ones shared above in the organization's mission statements, hint at the hurdles that internalized cultural values and thoughts appear to be imposing on the organization's business aspirations. In other words, the Agreement Cultural Model was viewed as interfering with and diminishing the organization's business aspirations.

Reason for the Model's Existence. Hinds et. al., argue that the actions usually draw on rules of behavior, which are typically culturally oriented [65]. Shore argues that these rules of behavior reside in the mind and in the world in the form of the cultural models. Here, I consider some rules of behavior that may be governing this cultural model of agreement.

The first reason seems to reside in the GSE business model itself. The model is based on the premise of getting "the maximum work done at offshore." Hence, the members of the GSE constantly try to get more and more work offshore. However, I speculate that while concentrating on getting the work offshore, they may not be in a position to always gauge its feasibility and required effort correctly because of the lack of complete information at the initial phase. Nonetheless, the best practice that the Indian vendor teams seem to have adopted is to say yes then and figure out the details later. Also, in the constant desire to impress the clients with their potentials and capabilities, they may by disregarding the long-term consequences.

The next reason hints at the Indian society's ties to cultural values that guide social behaviors around treating guests—an outsider to a community who may hold a higher status than the host himself. In this sense of the term, the vendor teams considered the client teams as guests. A popular Sanskrit verse reads as "*Atithi Devo Bhav*," which translates to English as "the guest is the God." The Indian teams treated their guests—the clients—very

specially. For instance, during the clients' offshore visit, the Indian teams had decorated the office's entrance with a traditional design from flower petals and the client's name was also written with flower petals under the design (represented as a placeholder) as shown in the Figure 11. Such decorations are commonly displayed at the entrances of (1) religious places (e.g., temples) and (2) significant locations in the house during certain auspicious occasions (e.g., door entrance to a house during festivals such as Diwali) to give a warm welcome to the guests. To treat the guest like a God may indicate that the host will make every effort to satisfy the guest's requests. The vendor teams appear to have internalized this cultural knowledge and their actions might be a manifestation of this cultural knowledge.

Figure 11: Client Team's Welcome Decoration with Flower Petals.

The last reason for this behavior may be explained with Hofstede's dimension of power distance and individualism/collectivisn, which he states are intuitively correlated with each other [68, pg. 102]. For instance, Hofstede reports that the cultures with higher power distance tend to accept and expect authority and collectivistic cultures tend to agree to group decisions over individual decisions. According to Hofstede, Indians possess the characteristics of a collectivistic culture, where maintaining social harmony and interpersonal conformity is important and confrontations are not encouraged [68]. Previous work on understanding culture in global software settings have viewed power distance and collectivism

to be the basis for the vendor team conforming to their clients' requests (e.g., [31]).

6.2 Trust Model

"Trust as a phenomenon is complex and has many meanings, and no widely acknowledged definition of the term exists" [73, pg. 17]. Trust emphasizes the notions of confidence, faith, and hope in the honesty and integrity of someone or something. For the purpose of my dissertation research, trust refers to the extent to which one cultural group believes in another entity (e.g., cultural group, system, or individual) with which they have an association.

Trust is one of the extensively researched topics in a global software engineering context (e.g., [76, 8, 7]). In fact, it has been considered as one of the fundamental factors that determine the failure or success of any GSE projects [97]. Past research has discussed various aspects associated with trust, such as investigating the impact of cultural surprises on trust elements [8] and illustrating the influence of trust on virtual teams [31]. I found some interesting dynamics of trust's association with testing through my studies.

Trust and Testing. The analysis of my data has led me to believe that trust plays a unique role in the GST practice. Testing-related parameters, such as how much testing is required, who will perform testing of each component of the system, and when the testing will be performed, are all governed by the extent to which clients trust their offshore partners, as well as the end-users of the system. For instance, the findings from the comparative analysis of Japanese Client (S3) Team and U.S. Client (S3) Team in Study S3, which I discussed in Section 5.1.1.5, provide evidence suggesting that the Japanese Client (S3) Team did not trust others to perform their testing activities. For another example, the Japanese Client (S3) Team team's hesitance to trust the way their users will potentially use the system led them to test the system exhaustively. As yet another instance, the Parent (U.S.-Mexico) organization in Study S3, adopted the practice of testing any new functionality at the client side first. These insights that emerged from the study analysis intrigued me to investigate

this thread further. I present the details of this analysis in Chapter 7.

Instituted Trust Model. In Section 4.1.4, I discussed the custom of capturing screenshots of every testing step that the vendor teams executed. These screenshots served as proof that the vendor teams performed their testing offshore. In other words, these screenshots served as publicly-available, objectified agents or mediums that represented and facilitated trust building and management. In this sense, these screenshots may be viewed as what Shore refers to as the instituted models of trust because they are the "externalization in the social world of particular models of experience" [124, pg. 51]. When the screenshots are considered independently, they might not be adding any business value to the specific activity of testing. In fact, in isolation, the screenshot-capture procedure and the screenshot output themselves may be viewed as another unproductive-productivity experience; the test engineers are being productive because they are producing the output artifacts (i.e., screenshots), despite being unproductive because these artifacts add no business value. However, when viewed in the entirety of the system, the screenshots indeed make an invaluable contribution because they form the externalized schematization that mediates the client's trust. This idea is distinctly illustrated in the following account, which occurred post-escalation, that manager S3T2.PM2 shared with me.
(S3T2.PM2)

We lost one whole week and what happens in that when something goes wrong... we will lose the confidence. What will they [client] do? They will [would] like to check everything, right? Now they started checking everything, whether these people [the vendor team members] are doing what they are saying... They got a doubt that either they are doing some cheating or they are just... You know how come if you are not able to do it [testing] in six days and now in two or three days you are able to do it.

So I am a human mind, I am going to doubt what you are doing. So what they did is they started doing the audit: 'I [client] want to check what execution you [vendor] have done till now.' Unfortunately we got one or two issues. See it happens, your 80% is accurate but 20% there will be issues, issues in the sense we have not done cheating but we have done fast runs... in the sense that without recording the proof of testing [screenshot], I have just done passed.

When you say that I executed the test case, the status is that it has passed... but at the same time I am supposed to attach the screenshot, write on the details that this is what I have done and this is the proof of doing that... So one of the guy[s] has [had] skipped [omitted] that step [of capturing the screenshots], though he has [had] actually executed the test case he skipped [omitted] that step.

(Quote 30)

The above vignettes describes how the trust was at jeopardy: "when something goes wrong... we will lose the confidence." When there is a lack of trust ("doing some cheating") one consequent action is increased monitoring, which the client team implemented in the form of "checking everything" and "audits" [97]. Unfortunately, the increased monitoring uncovered some loopholes in the testing conducted by the vendor teams—"fast runs"— where the screenshots were omitted, distinctly highlighting the screenshots' role in the manifestation of trust. However, the fact that the screenshot-capture action was the only action chosen for omission may imply that the action manifested the characteristics of generating the *unproductive-productivity* experience. Needing to decide whether to omit the screenshot-capture action occurred because of the high deadline pressure situation, juxtaposed with the client-related raised tension; I refer to such situations as quality-dilemma situations, which I discussed in Section 4.1.5. It appears that the test engineers made a conscious choice to compromise quality to meet the scheduled deadline. As a result, the

importance shifted from capturing the screenshots to delivering the promised test executions on time. It is evident in the participant's explanation above that the vendor team did not comprehend the culturally-embedded meaning of the screenshots and hence they seem to have chosen to compromise the quality to finish the task by executing the "fast runs." However, it is worth emphasizing that the screenshots were indeed crucial from a cultural viewpoint—for the trust building process—because they were the embodiment of the vendor team's credibility.

Vendor Teams: Ignoring Credibility Concerns. In the Section 5.2, I narrated the escalation issue with its reasons and consequences. One distinct consequence of the escalation account was that, in the eye of the Mexican Client (S3) Team, the Mexican Vendor (S3T2) Team's credibility took a nose dive. However, I observed that the vendor teams skimmed over the credibility issues but did not give them the importance they deserved. I believe that the reasons for this ignorance were rooted in some basic belief systems that the Mexican Vendor (S3T2) Team had internalized.

The project management members of Study S3 thought that they did not lose their credibility because they worked overtime and had finally finished the tasks before the deadline. Thus, their perception was that as long as the tasks assigned were completed, escalation accounts, such as the one narrated in Section 5.2, had no implications on their credibility.

Interestingly, the members at the lower level in the organizational-structure hierarchy acknowledged that trust was an issue. However, their tone indicated that it was the client who lacked trust in the vendor teams. They did not seem to analyze the cause of the distrust or acknowledge the actions that consequently led to (or enhanced) this distrust. Thus, the vendor teams did not realize the profound impact of ignoring credibility issues and, as a result, they did not give credibility the due importance it deserved. Interestingly, the U.S. Client (S3) Team managers seem to have observed this behavior of the U.S. Vendor (S3T1) Team in the past, and so they had added feedback in the annual review of the

partnership mentioning them to be realistic in their commitments because "credibility is [was] extremely important."

Thus, the insights that emerged from the study analysis suggest that the Trust Cultural Model resides on the client-vendor periphery and forms the foundational basis for determining technicalities associated with GST's realization and maintenance of a functional relationship between the client and the vendor.

6.3 Flexibility Model

The Flexibility Cultural Model refers to the degree to which one culture is willing to adjust to the environment and situation at hand. This definition does not necessarily consider the degree of adjustment needed to fit in or collaborate with another culture, but includes the degree of adjustment used towards achieving a goal. At the vendor organizations I studied, I found that the participants demonstrated flexibility on various aspects—work timing, work duration, frequently changing work priorities, varying degree of work pressure, and changes in testing tasks, scope, and plans at a short notice.

Based on the study insights, it is evident that in these Indian vendor organizations it was taken-for-granted that the test engineers would be willing to work overtime and be flexible about their work hours and schedule. My personal experience is an exemplar of this expectation: when I was performing the legal formalities to conduct my study at Vendor V2, I was asked to sign a form (signed by all other new employees joining the organization) that stated that I would be willing to work flexible hours if the project required it. Such forms and other artifacts constitute the "instituted models" of the flexibility cultural knowledge because of their publicly available objectification.

When project deadlines approached, it was assumed that the engineers would put in whatever time was required to get the work done, including sacrificing their pre-approved vacation times if needed. Interestingly, female participants, from each of my three studies, expressed their preference to work in testing projects as opposed to development projects

because the frequency of working overtime and over the weekend was lower in testing projects than in development projects. Thus, by working on testing projects, the female participants reduced the number of times they had to compromise their family commitments because of work demands.

Some work has been done in the management literature that discusses a concept similar to Flexibility—Adaptability. In fact, "*adaptation* refers to the changes that workers make to their work practice," with convergence occurring when practices become similar over time and divergence occurring when practices start differing over time [65, pg. 160]. Crampton and Hinds further expand on the idea of adaptation by proposing an embedded model of cultural adaptation. However, this concept of adaptability differs from my concept of flexibility in terms of the ideas that the two concepts emphasize. On the one hand, adaptation emphasizes the inter-cultural adjustment and it refers to the changes in individuals' behaviors and thoughts, and efforts invested in resolving contradictions between impinging cultural system. On the other hand, flexibility emphasizes the extent to which an individual demonstrates adjustment or elasticity towards a work-related situation or problem at hand. Flexibility does not necessarily refer to inter-cultural adaptability. These two concepts may be viewed as complements to each other, with adaptation residing on inter-cultural boundaries and flexibility residing within a cultural system. To illustrate the difference, consider an American who is not willing to work on weekends on a regular basis (i.e., does not demonstrate flexibility), but he bows in front of his Japanese colleagues to greet them with respect (i.e., demonstrates adaptability).

Hinds et.al., describe that different cultures vary in their extent to be adaptable, thus, implying that "culture may matter for adaptation" [65]. I recorded similar insights with respect to flexibility based on my study. I found that the Mexican (S3T2) Onsite Coordinator Team demonstrated a lesser degree of flexibility compared to their offshore counterparts. The onsite members were reluctant to work at later hours or be flexible about their availability beyond work hours. To compensate for this reluctance at the onsite location, the

offshore team adjusted to the working styles of their onsite team. Participant S3T2.TM1 shared an experience of one such adjustment: his daily practice was to call his onsite coordinator while he commuted to and from work so that he could perform the required "handshaking" by updating each other on the project's progress during the work hours of his Mexican (S3T2) Onsite Coordinator Team.

Additionally, Hinds et.al., discuss various cultural facets, such as power dynamics between two locations, that appear to be crucial ingredients for understanding the extent to which adaptability will be provided by either one of the locations [65]. However, my findings suggest that power dynamics were significantly less influential on the cultural facet of flexibility than the cross-cultural facet of adaptability. In fact, if power dynamics were playing a significant role, then I expect that the offshore vendor team would have held a more dominant position than their onsite Mexican colleagues for the following three reasons. First, both the teams worked for the vendor organization, unlike the client-vendor setting where the client team holds an evidently dominant position. Second, Mexico and India, as nations, are both developing economies, which suggest that economic powers may not be acting as differentiating factor. Third, the vendor organization was based in India, which implies that Indian cultural values and practices might be dominant in the organization, making the Indian team the dominant group. Despite these reasons that suggest that the Mexican onsite coordinators were not holding a dominant position in the team's configuration, the offshore team were the ones who accepted and adjusted to their onsite team's style of working. The offshore delivery management level manager (S3.DM2) explained that "we [Indian offshore vendor team] are used to the global delivery model, we are used to our people being deputed to work in different countries to work there, understand[ing] our expectations, working at night, and getting the things done. But when we recruit people in Mexico they are guided by the laws of the[ir] land."

This illustration orchestrates the following three insights. First, the Indian vendor teams have fully internalized the Flexibility Cultural Model while the Mexican onsite team did

not demonstrate internalization of the model to such an extent. Second, the "global delivery model" practice (which I describe in detail in Section 6.4) is an embodiment of flexibility-related cultural knowledge (i.e., Flexibility Cultural Model). In other words, this business practice is governed by the cultural systems such as the Flexibility Cultural Model and, consequently, any change in the cultural systems will have a ripple effect on business practice (I expand this discussion with an illustration from the study in Section 6.4 and Section 6.5). Finally, the Flexibility Cultural Model may be providing the foundational model on which India's "global delivery model" has been evolving, thriving, and "surviving."

The "Survival" Model. ACM's detailed report on "Globalization and Offshoring of Software" states that the inception of global software-services' popularity began from two countries—Israel (1960s) and Ireland (1990s) [12]. However, there was a revolutionary change in the late 1990s when India entered the global software-services market. Soon, India emerged as the single most important and largest provider of global (offshored) software services [12, pg. 109]. Despite other countries being pioneers, it was interesting to note India's role as the emergent leader. The report listed various reasons for India's popularity in this market including education (i.e., emphasis on mathematics and science), economy (low wages), lack of skilled resources, and English-language capabilities. These reasons, which represent cultural facets or culturally-influenced facets, strongly suggest that culture plays a significant role in India's success in this market.

Interestingly, the practitioners of this global software business (i.e., my participants) had an additional intriguing explanation for India's success. A senior manager at the delivery management level (S3.DM2) expressed his thoughts on India's success by using the conceptual analogy of "lethargy." He explained one reason that the work started coming to India and China[1] was because the U.S. was "lethargic" as they were "not ready to

[1]The ACM report states that the kind of jobs that came to India and China were fundamentally different. While the business that came to India has focused on exporting IT and software services, Chinese business has focused on localizing the software and products for the Chinese market [12, pg. 121].

work for more than eight hours." The manager seemed to suggest that the internalized Flexibility Cultural Model was important for the Indian vendor organizations to thrive at this outsourcing, offshoring business. Not surprisingly, another project management level participant (S3T2.PM2) had expressed similar understanding and described the Flexibility Cultural Model as the "survival" model that helps them thrive in this business.

(S3T2.PM2)
. . . we are not losing [the client's business]. We are somehow surviving, surviving, surviving because our team is ready to put that extra effort, we are really committed. . . the consequence is that we may burn out eventually. That is a right consequence but losing the credibility from [of the] customer, this does not happen. (Quote 31)

In fact, the vendor teams considerably relied on the Flexibility Cultural Model in many situations. For instance, as depicted in the illustration above (also refer to Section 6.2 for further discussion), the offshore vendor teams had internalized the notion that as long as they worked overtime and finished the promised job, there would be no repercussions in terms of jeopardizing the client's trust—"losing the credibility from [of the] customer, this does not happens." Moreover, some participants acknowledged that, despite the "survival" and rescue support that the flexibility practice was rendering, excessive reliance on such cultural practices resulted in "suffer[ing]" because there were raised client expectations related to work output and productivity.

(S3T2.PM2)
. . . and then we suffer but what is happening is we deliver eventually because of the Indian way of working. We owe them [the client team]. We show our flexibility, we call people in [at] night, seriously, all the time. (Quote 32)

Thus, this vignette manifests the idea that the cultural systems embedded in the "Indian way of working" allows the vendor teams to "deliver eventually" by "showing our [their] flexibility." In other words, these thoughts suggest the vendor teams' reliance on such cultural systems to accomplish their business commitments and seek refuge from complicated cross-cultural conflicts (e.g., losing client's confidence).

6.4 Culturally-Embedded Global Software Delivery Business Model

In this section, I discuss the meaning of the global software delivery (GSD) model or the "global delivery model" as the participants commonly referred to it. This model has been the business model adopted by all my participant organizations and it has been governing the functioning of the various GSE practices, including the GST practice that I studied. This section illustrates how the GSD business model was deeply culturally rooted. I start with the discussion of the GSD model's definition as adopted by the (industrial) professional organizations (Section 6.4.1). I then present the hidden cultural dimensions of the GSD model (Section 6.4.2) that are tacit and hence are often unnoticed. However, they are crucial in shaping the GSD model's practice and determining the success of this model in any setting.

6.4.1 Idealized Definition of GSD Model

Global software development is commonly referred to as Global Software Delivery (GSD) in the software-services sector of the Indian Information Technology society. It refers to the concept of providing software-development services to the clients from across the globe. The idea that forms the basis of this model's concept is that the practitioners will perform software-development related activities and provide delivery 24x7 across the globe. As described by one participant
(S3.DM2)

Global delivery model is nothing but the concept that you might be sitting in U.S., I will get your work done in a global delivery model way like [such as], in the U.S. hours, U.S. people will be working and when you U.S. goes to sleep, India wakes up and India will start working, and then we sleep you [U.S.] start working. So this is the global delivery model. So, it is almost like 24 hours of the day we will be churning out work no matter where in the globe. We can go from U.S., then go to U.K. with the time gap of six hours, then come to India with the time gap of five hours, then we can go to Mexico with the time gap of twelve hours, and then go to U.S.. That is global delivery model, delivering from anywhere in the globe. 24 hours churn out. But that requires a lot of coordination. That handshake has to happen. U.S. to U.K., U.K. to India, India to Latin America, Latin America back to U.S.. So, if that model has to succeed, it has to be a handshaking model. (Quote 33)

This approach of "churning out work no matter where in the globe" is commonly referred to as the follow-the-sun approach and its a popular approach that the top software service-provider organizations adopt. In fact, the professional organizations define their GSD practices referring to the follow-the-sun approaches. Their definitions primarily emphasize two parameters— time and geography. Time is represented in the form of work being done 24x7 and geography is represented in the distribution of the work across the globe.

6.4.2 Cultural Facets of the GSD Business Model

The participants and practicing organizations define GSD model emphasizing on the two dimensions—time and geography. However, my analysis revealed other interesting dimensions of the GSD model that are not distinctly visible in these definitions. These hidden dimensions resided in the form of tacit knowledge internalized by the practitioners and reshaped the meaning of the GST practice. However, these hidden dimensions do play a crucial role in governing the success of any projects that adopt the GSD model. Moreover, as I will describe in this and the next chapters, neglecting these dimensions resulted in almost failure of one of the project that I investigated. In this section, I describe the underlying meaning of GSD practice by presenting a discussion of the hidden dimensions of the GSD model.

6.4.2.1 *Team Structure: Onsite Coordinators*

Although the GSD model's definition does not provide any description of the team structure, the study revealed that the team structure emerged to be one important component of the model. The basic GSD model has a commonly accepted team structure as represented in the diagram below:

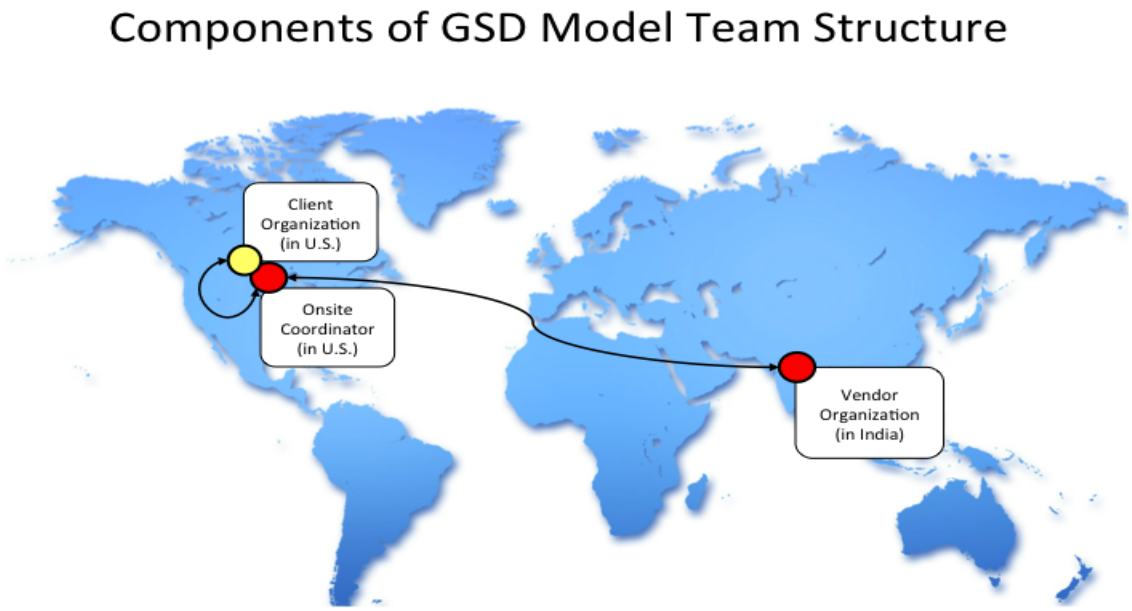


Figure 12: Components of the GSD Model's Team Structure

There are three components (Figure 12) of the team structure in the GSD model. First, the offshore vendor team, who is the service provider. Second, the onsite-coordinators, who are vendor team members, but reside close to the clients; their primary duty is to provide coordination support to the offshore team back in the vendor organization's location. Lastly, the client team who is requesting the service from the vendor team. This is the basic model of GSD recognized widely.

However, the study insights revealed that there are hidden elements of the team structure that emerged to be crucial for the success of the GSD model. One such element is the

cultural background of the onsite coordinators. Although the onsite coordinators belong to the same vendor organization, hence shared the same organizational culture with their fellow offshore team members, their nationality (or extent of acculturation to other cultures) played a critical role in the hassle-free functioning of the GST practice.

In most of the GSD adopted GST settings, the onsite coordinators in the Indian software-service provider organizations are Indians, who are deployed to various places around the globe for performing the task of coordination. This configuration is depicted in the figure below.

However, there was a lack of awareness of the deeper impact that this particular configuration held and, hence, there was no acknowledgment given to the importance of this particular parameter in regards to the team configuration setting for GSD. Nonetheless, the significance of the cultural configuration surfaced when this configuration was changed. I found that in the Mexico project, they had a unique team configuration—the team included the onsite coordinators, who were Mexican nationals residing in Mexico and working closely with the Mexican clients—instead of the typical configuration which includes Indian nationals as the onsite coordinators.

The study insights provide evidence that suggest that this change, although not perceived as a threat by the team, had apparently caused considerable difficulties for the offshore vendor team in their execution of the GST activities, which I discuss in detail in the next section.

6.4.2.2 Failure: First Time Failure Is Common

The GSD model, although portrayed as a static business model, is indeed a dynamically evolving relationship model of the vendor and the client organization and, more importantly, the respective involved teams. The model seems to layout a learning opportunity for the teams (and organizations) to acquire understanding of their counter team members at different levels including working style, cultural practices, and organizational inclinations.

Because it is an evolving model, the initial iterations of learning may be challenging and success may not always be bound to happen. In other words, initial failures when new engagements are established between vendors and clients are quite common.

However, the model does not highlight this knowledge of first time failure being common in any direct or indirect fashion. I found that the experienced GSD practitioners (mostly vendors) do not find surprising the possibility that a new project with a new client may fail the first time or things may not go as planned for the first time in the testing life cycle. Particularly, it was more accepted by the practitioners when it was their first project with a new client from a relatively newer geography with whom they have not worked in the past.

Interestingly, my the Mexican client project that I studied, had experienced such a situation. The Mexican Vendor (S3T2) Team had viewed their failure to complete the testing tasks as planned in the cycle one of testing as an acceptable common occurrence. Hence, they were not anxious of such a failure or anticipated any threat to their relationship with the clients. However, for the Mexican Client (S3) Team, this seemed to be a sign of inability or lack of capability of the vendor organization because they interpreted that the vendor organization had failed to create a positive first impression. This resulted in credibility failures issues that are discussed later the Section 6.2.

6.4.2.3 Duration: Working Extended Hours

In the participant's description of GSD model presented earlier in this chapter, the participant presents this concept as "churning out work, no matter where in the globe." Furthermore, he mentions that for the success of this model, there ought to be handshaking done among the distributed teams.

Because of the timezone differences, many times these handshakes need to take place at odd hours (e.g., early in the mornings or late in the evenings). Being available at the time of the clients' (for the offshore team) or vendors' (for the onsite coordinators) work hours

means that the practitioners have to work for extended hours to facilitate this handshaking. In fact, I found in all the studies that the timezone differences, most of the time, were covered by the Indian offshore team members. This meant that the offshore team members usually worked for extended hours on a daily routine basis. My participants were flexible to meet early in the mornings (sometimes when they were driving to work) or late in the evenings or often both to accommodate the convenience of the clients or onsite coordinators.

I found that the Indian team members—onsite coordinators and the offshore vendor members—were accustomed to this "handshaking" model in terms of the work hours and duration. However, there was significant hesitation on the Mexican onsite coordinators part to accept this handshaking model, which required them to work beyond 8 hours and invited challenges for the teams. Thus, based on the study findings, it is evident that the GSD model strongly relies on the handshaking model for its survival, which in turn heavily depends on the flexibility model (as discussed in the Section 6.3).

6.4.2.4 Expertise: On-Job Learning Model

A common practice in the GSD model for adding and training new members on the project is to bring in recent college graduates and allot them activities on the project such as executing test cases or detailing out the steps in test-cases creation. Although the clients are told that the offshore vendor team has all experts on their team, the reality is that some of the team members would be recent college graduates with little or no experience working on real-world projects. One reason behind not disclosing the novice on the team is when the vendor teams negotiate to earn the project from the clients, they portray an image that they have many experts in this area. However, while they do have experts in the project's area, they also plan to use the client projects as a on-job training opportunity to train novices to become the new experts, who can then be showcased for other clients in similar technical areas.

Thus, this information—of the team being composed of a mix of experts and novices test engineers—is a classic example of the cultural knowledge internalized by the Indian GST practitioners, which may or may not be necessarily known to their respective client teams.

(S3T2.PM2)
. . . whatever we show on the presentation 100% is pakka *[Hindi word that means certain], as if everybody is an expert, right? Out of 100 people 40% are expert, 60% are fresher, trainee guys so they are going to do [commit] mistakes. . . but it is our model, we deliver it [the outcome]. We can't help it.* (Quote 34)

6.4.2.5 Planning: Curved Projection Test Planning

Based on the experience providing GST service to many clients, one feature that has become a part of practitioners is the "curved-projection" test planning. This kind of test planning refers to the planning where the teams plan for reduced amount of testing in the initial weeks—when they speculate issues such as environment instability and infrastructure unavailability—and increase the testing scope in the later weeks.

However, I found that this piece of knowledge formed a part of the tacit knowledge of the more experienced vendor-side GST practitioners (i.e., my participants), which was not necessarily acknowledged, expressed or explicitly shared with the less experienced client teams. However, it was evident that this curved projection of test planning was indeed an important element for the success of the testing project:

(S3T2.PM2)
For (test) planning . . . we always plan 10-15% in the first week, 20% in the second week, slowly gradually increase, so by that time the environment stabilizes the data issue is resolved, so we get the speed. It should be a curve; it cannot be a flat projection. We tried to do that curve projection but the customer pushed for this flat projection. They said "no we need everything" and we are not able to convince them. (Quote 35)

The participant's (S3T2.PM2) use of the term "always" implies that the curve project was their commonly adopted and preferred test planning strategy. However, this strategy resided within the vendor teams cultural system of practice, thus failing to reach the client.

6.5 Unfolding the Escalation Situation Through a Cultural Lens

In this section, I revisit the reasons for the escalations and explain the reasons by unfolding the scenario through a cultural lens—illustrating how these reasons were influenced by different cultural elements.

6.5.1 Internalization of Cultural Models

As described earlier in Section 6.4, the typical team structure in the global software delivery model include an onsite coordinator who is from the vendor organization but is located at the client's site. In the typical global software delivery setting, the onsite coordinators belong to the same cultural group as the vendor organization. However, in the Mexican client-vendor project, the onsite coordinators were "local hires" from Mexico, whereas the offshore vendor team consisted of Indian team members.

Although this was not seen as a significant change in the setting of the GSD model, the change was considerably significant when viewed through a cultural lens. The onsite coordinators now no longer shared the same cultural knowledge (i.e., the cultural models) as their offshore counter parts. For instance, the Mexican onsite coordinator team was new to the GSD business model. Hence, they were unaware of the various hidden cultural dimensions of this business model. The Mexican onsite coordinators showed reluctance and resistance to working at odd hours and for extended durations, which indeed was a crucial factor for obstacle-free "handshaking" between the onsite and offshore teams. This reluctance created hurdles such as schedule delays, which resulted in the escalation. Thus, the Mexican onsite coordinators had not embodied the same meaning of the *duration* dimension that was emerging to be a crucial factor for the GSD model success. In other words, they had not acknowledged the crucial role of the Flexibility Cultural Model (Section 6.3)—a prominent ingredient for the GST practice's recipe for success at the studied Indian organizations.

The Figure 13 depicts the traditional GSD team's structure with onsite coordinators

from India and Figure 14 depicts the new Mexican client-vendor team's structure with onsite coordinators from Mexico. The figure shows that as compared to the traditional setting, where it was assumed that the onsite coordinators understood the implied meaning of the embodied cultural models, the Mexican teams did not embody the cultural meanings.

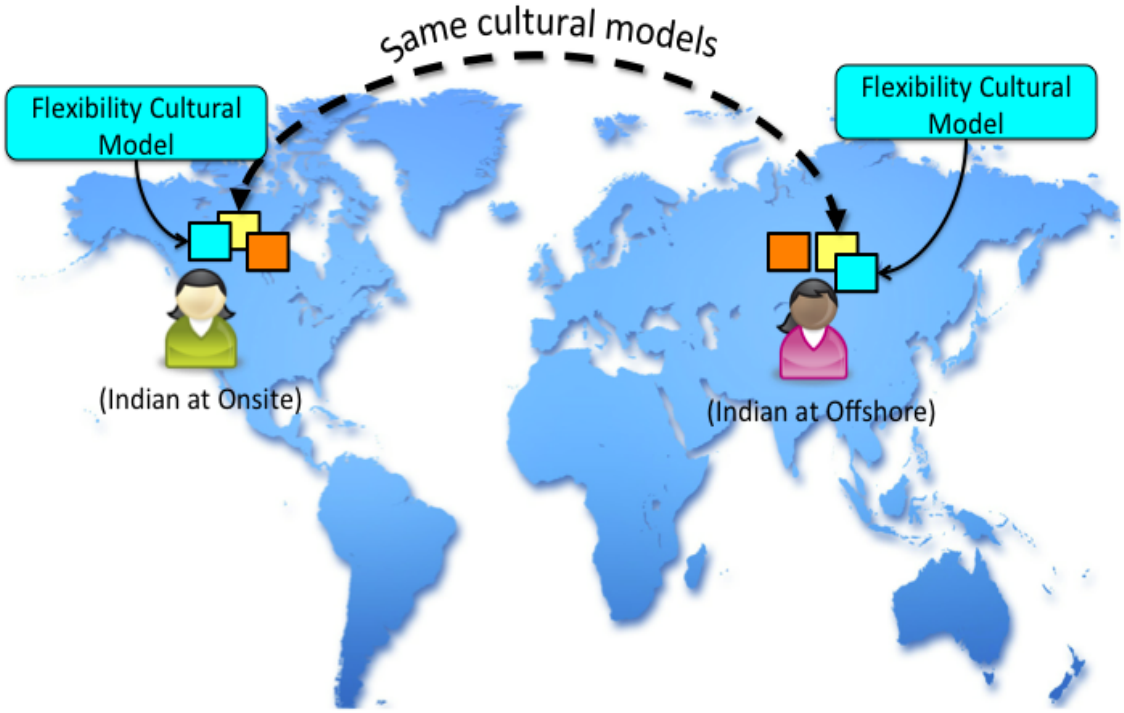


Figure 13: The Traditional GSE Model Team Structure

6.5.2 Interplay of Cultural Models

Study evidence suggests that there was a general agreement among members of the Vendor V3 organization (I also observed this in other participant organizations) that the escalation story was a common experience for the vendor team members. The clients ask for the testing task to be completed in a given amount of time. The offshore team is certain that they would not be able to accomplish this task, but they said yes to it anyway. Not surprisingly, the offshore team failed to accomplish the goal. This resulted in the clients losing trust in the capabilities of the offshore team and in turn the organization. To accomplish the goals, the vendor organization relies on their flexible working styles and completes the set goals

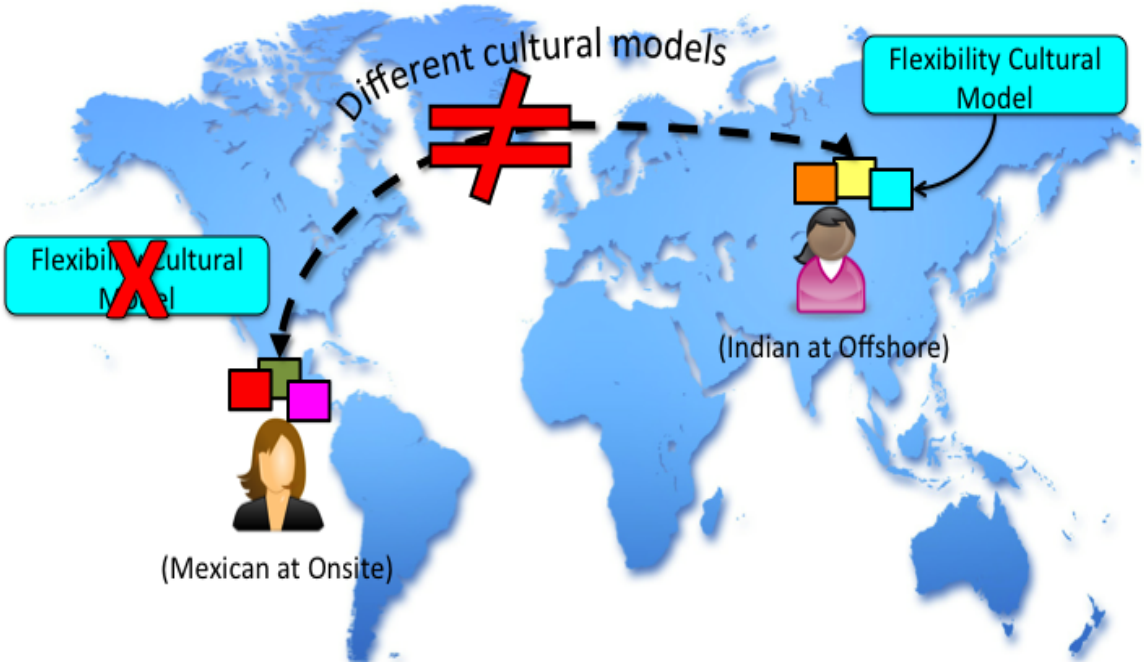


Figure 14: New GSE Model Team Structure

by working extra hours and over weekends.

Now that the visible aspect of this episode is known, lets view it again through a cultural lens. Figure 15 illustrates the episode in a cultural milieu. When the team said yes to the clients, they reacted based on the Agreement Cultural Model that they have internalized — which refers to the degree to which one culture agrees with others. When the clients lost trust in the vendor organization, the trust cultural model was highlighted — which refers to the extent to which one culture believes in others. Finally, the organization relied on the Flexibility Cultural Model to accomplish the set goals, which refers to the degree to which one culture adapts and adjusts to given situations at hand.

However, in this process, the vendor team did not seem to believe or realize that they were jeopardizing their own credibility. Moreover, under the extreme pressure situation that was created at the time, the vendor team members appeared to have faced a quality dilemma situation (discussed in Section 4.1.5), which led to quality compromises unknowingly or knowingly further hampering the trust of the clients.

Thus, the internalized agreement cultural model appeared to have landed the vendor team into trouble, which they admitted during the interviews (e.g., Section 5.2.2.4). However, when I engaged some participants in the discussions around this models and their tacit behaviors, they quickly changed their tune and began becoming aware and speaking about it. In fact participant S3T2.PM2 described that " we as Indians will never be able to say 'No' to the customer [client]" because this behavior is " in our [their] Indian genes."

Interestingly, despite knowing that agreeing to the client's planning strategy would most likely lead them to failure, the Mexican Vendor (S3T2) Team proceeded in that direction because they appear to implicitly rely heavily of the Flexibility Cultural Model for their rescue and "survival." They tend to take the risk of pleasing the clients, by following the client's desires, thinking that if the plan does not execute as desired, they would come and work over the weekends and make up for the delay in the schedule. However, in this particular situation, the making up over weekend plan backfired at them, raising more doubts in the minds of the clients—"They got a doubt that either they are doing some cheating or they are just... You know how come if you are not able to do it [testing] in six days and now in two or three days you are able to do it" (Quote 30). However, ironically, the Flexibility Cultural Model still was the strong relying factor that the Indian vendor teams resorted to fix the damages done by their enactment based on the Agreement Cultural Model. The interesting element of this situation's enactment was the vendor managers' beliefs that the credibility was not hurt—"Some dips [mistakes] may have happened, you have to accept the fact. But we have delivered, so that is [has] not resulted in... we losing the [client's] credibility [trust].(S3T2.PM2)"

The Flexibility Cultural Model seems to be a crucial cultural system element that drives the success of global software delivery practice in India. The teams considerably rely on this model to overcome some of the most common challenges that are experienced in distributed teams — information exchange, timezone challenges, and language challenges.

For instance, in all the three studies I conducted, I found evidence that indicated the Indian vendor teams worked flexible hours and for extended duration to ensure handshaking occurred obstacle-free between the onsite and offshore teams. It was implied that when the project was slipping out of schedule, which occurred quite frequently, the team was expected to work extra hours. Weekend working was a common practice and participants who were reluctant to follow the practice were frowned upon. In fact, the managers sometimes took pride is showing how they fixed a situation by working extra hours and coping with the delayed schedule.

The fascinating element was that the GSD practitioners were unaware of this pattern of practice that they followed and their heavy reliance on the model, because they viewed their flexibility as a matter of pride. However, the study evidence suggest that their tacit reliance on this model probably became the driving factor to enact based on the Agreement Cultural Model. Thus, there was a lack of realization that their heavy reliance on the Flexibility Cultural Model was probably obstructing them from making calculated and informed decisions related to testing planning and scheduling. This might explain the reason why the Agreement Cultural Model dominated despite they experiences conflicting thoughts (explained in Section 6.1).

This episode of cultural interplays illustrate how the underlying cultural system dominates and maneuvers the technical system of GST practice.

6.5.3 Cultural Sensitivity Element Missing

Throughout the investigation of the escalation issue, the participants compared the Mexican team's experience with the U.S. team's prior experience. The participants mentioned that such escalations would not have happened with the U.S. Client (S3) Team. They accredited the success with U.S. Client (S3) Team to the client's "maturity" in the sense of allowing the U.S. Vendor (S3T1) Team to perform their testing activities their way. However, my study analysis revealed that the success was not only because of the "maturity" of the U.S.

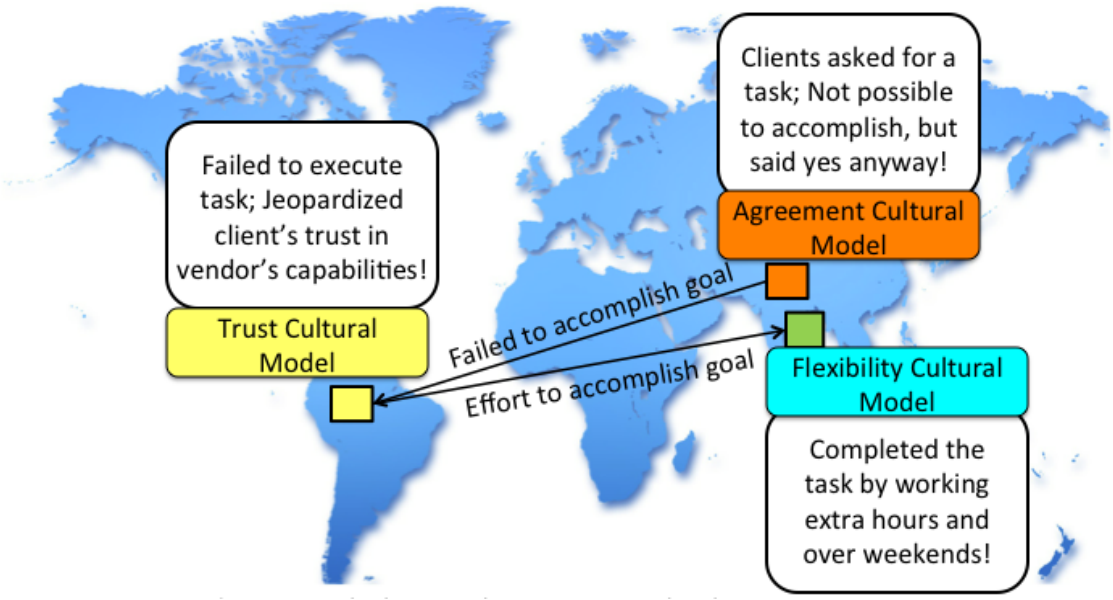


Figure 15: Depiction of Cultural Interplays

Client (S3) Team but also because of the cultural sensitivity of the U.S. Client (S3) Team team, which went unnoticed and unacknowledged by the U.S. Vendor (S3T1) Team team.

The U.S. Client (S3) Team, particularly the client project manager (S3.CO1), was aware and sensitive about the adopted Agreement Cultural Model that the Indian vendor team was unknowingly enacting. Hence, the client team was always vigilant when the U.S. Vendor (S3T1) Team team overcommitted or tried to commit unrealistically. The U.S. Client (S3) Team would emphasize that the U.S. Vendor (S3T1) Team "be realistic when committing," which meant that they should not plan for something that might not be doable in the given amount of time and that would require extra hours of working. In addition, the client project manager, during her visits to India (offshore location), had made an effort to meet and connect with the offshore team and get to know them and their work practice. She shared that such connections had helped her understand how the offshore team works, which in turn helped her understand how they should be planning their testing strategies:

> One of the things I learned was, like it was in one of my trips, I think, I was in Chandigarh at the time. . . I don't know how many years I've been working with

> you guys [vendor team members], maybe for over a year or two. I finally found out that most of the people who work here are a lot of the young ones and on weekends they go back to their families and they do their religious things. So, as a client, I never understood what I was really telling them when I said work on a weekend. Because in America, when we work on weekends, we can work from home, we log on, and there is not a big sacrifice. Maybe I miss a T.V. show, but in the India, when I ask to work on a weekend I am actually taking them away from their family time. They are going to see their family . . . that's the only time . . . because they are working incredible hours during the week. But, had I not had the accidental conversation I wouldn't have known about it.

Thus, the client manager not only put in effort to connect and get a sense of the work life of the teams offshore, but also attempted to understand the connection between the personal and work life of the participants. Additionally, some client managers put in efforts such as having small talks with the participants, keeping themselves updated and joking about the inter-team members' marriages that occurred offshore, and asking about the things the offshore team members wore and ate. In conclusion, I believe that a significant portion of U.S. project's success was embedded in the client manager's sensitiveness towards the Indian cultural practices.

However, interestingly, such a sensitivity was not evident in the Mexican Vendor (S3T2) Team. The paradox was that the vendor team could not even explain and make the client team aware of such sensitivity mainly because (1) they were themselves not aware of it many times and (2) given the team configurations, power distances elements, and the specific client-vendor relationship it would have been awkward for the vendor members to raise awareness about their own culture to the client teams. I speculate that the lack of such sensitivity on both the sides—the Mexican Client (S3) Teamand the Mexican Vendor (S3T2) Team—contributed to the escalation. The paradox is aptly explained in this conversation between the client and the vendor manager in one of the focus group sessions:

[Vendor Manager]: you [client manager] took that initiative to ask us, right? We, as a vendor team, would not say that 'they [vendor team members] cannot come because they don't have laptops there [in the offices], that desktops are not fixed, they don't have facilities to work from home.' I will not say this to you [the client] because I will think you will say that he is finding reasons to not get the work done on time but that is not the fact.

[Client Manager]: Yeah, but the sad part is if I knew that, I would maybe think about my decisions differently. But if I had no clue of what I am disrupting I wouldn't think about it. So sometimes you guys have to handle more than you really need to because I don't understand.

6.6 Consequences: Neglected Damages

There were two consequences of the escalation situation that resulted from this particular episode, which are discussed as follows:

Increased Monitoring and Hampered Business. Because of the sequence of events, the vendor team had lost the trust of their Mexican Client (S3) Team. As a result they had to put in extra effort and suffer the consequences because of the lack of trust. Interestingly, the vendor team was in turn thinking that the client team was rigid and imposing specific rules: "they have to tell us exactly 'no, you have to wait, let us do it and only then will you do it."' The vendor teams were accusing the Mexican Client (S3) Teamof being less trusting. The Mexican Vendor (S3T2) Team had neglected the factor that made the clients become so distrusting of the offshore team. The vendor managers mentioned that its common to be less trusting for new accounts (clients), but creating situations where the clients could further lose trust in the vendor teams' capabilities was not acknowledged. The analysis revealed that as a result the Mexican Client (S3) Teamhad started monitoring the vendor team's activities closely. Moreover, the manager thought that the escalation was just a one time event that had no

consequences on the client's trust. However, what the manager failed to realize was that the client teams had started taking some preventive and cautious measures, such as performing more testing at client side, as a result of this loss of credibility. This action was surprising to many client and vendor managers, because it was defeating the purpose of using the offshoring service and thus hampering their business.

Raised Expectations. Another outcome of the escalation issue was the exposure of the Flexibility Cultural Model to the client team. The vendor team members reported that the Mexican Client (S3) Teamhad started taking advantage of the Flexibility Cultural Model of the vendor team. One onsite coordinator (S3.OC1) reported that the Mexican Client (S3) Teamwas, in a sense, trying to "exploit" the Mexican Vendor (S3T2) Team because they had now realized that the vendor team was significantly flexible in their working styles. Moreover, they would be adamant in planning testing activities based on their convenience (e.g., releasing product costing related information on a Friday) and expected the vendor teams to accommodate and plan their schedules accordingly. As a result, the vendor teams frequently ended up working for extra hours or over weekends. The onsite coordinator (S3.OC1) mentioned:"One example I can give to you is that [vendor organization] team has a whole week to complete the execution plan, but client put[s] a deadline of having everything finished on Wednesday and that offshore team should be working all weekends to meet this approach [deadline]."

CHAPTER VII

COMPLEX CULTURAL SYSTEM OF TRUST AND TESTING

7.1 Characteristics of Trust

In Chapter 6, I defined and described the cultural models that are embedded in the GSE practice. The Trust Cultural Model emerged as one of the vital cultural models embedded in the GSE practice. The Trust Cultural Model possesses several characteristics that play a vital role in the way trust dynamics unpack in the GSE setting. First, trust is related to *expectation*. Al-Ani defines trust as "one individual's expectations of another, and the former's willingness to be vulnerable based on those expectations" [8]. Some expectations are implicit and others are explicit. When someone has trust in another entity they have a sense of confidence that the other entity will behave as expected. Thus, meeting expectations raises trust levels and vice versa. Second, *trust and distrust can co-exist*. Dimoka illustrated that the brain internalizes trust and distrust separately because they are distinct and not two different ends of the same spectrum [47]. While trust evolves slowly and gradually through mutual interaction with the opposing party involved in the collaboration, distrust is highly sensitive and a small change or diversion from the normal expectation can result in catastrophic outcomes. Lastly, trust is *dynamic and highly fragile*. Considerable research has been done in the past to investigate the dynamic and sensitive nature of trust and its impact on the organizational practices, such as the GSE practice that I studied. Al-Ani and others have described the dynamic process of trust in the form of different stages, namely formation, adjustment, dissolution, and restoration [6]. Moe and Šmite investigated key factors that disrupt the fragile relationship of trust and the disruptions' effect on the software projects that they investigated [97].

These and other research investigations shed some light on the complexity involved in

accomplishing trust in the GSE setting. One reason for this complex system of trust is that the GSE practice is formed by intricate team configurations that span across different layers of cultural systems—occupational, inter-organizational, and cross-cultural teams. Synthesizing the different elements and their interactions in one unified environment of the GSE system gives rise to newer trust-related challenges. Although past research has individually investigated some of these cultural layers individually, there is a need to paint a complete picture of these cultural layers and their interplay within the context of trust and testing. Hence, in this chapter, I present the discussion of my analysis that focused on investigating the complex cultural system of trust and testing. I start with the description of a novel perspective on the internalizing of the testing process as an embodiment of the trust accomplishment mechanism rather than the commonly viewed notion of quality assurance (Section 7.2). Next, I discuss the three cultural layers that impose challenges to trust accomplishment goals in the GST setting (Section 7.3). Finally, I present a discussion of the reasons for trust violation and failure in the GST settings (Section 7.4).

7.2 Testing: An Embodiment of Trust Accomplishment

The literature on trust seems to have focused on the meaning and description of trust more with respect to the idea of having human beings as the participating members—trustor and trustee—and less with respect to the idea of having other entities (e.g., software system) as the participating members.

Interestingly, in the software-engineering practice discipline, testing involves ensuring that the system behaves as expected and the system does not behave as not expected. Thus, testing can be viewed as a process of gaining the confidence that the system will function as intended and that it will not function as not intended. Beizer states that "testing, pass or fail, reduces our perception of risk about a software product. The more we test, the more we test with harsh tests, the more confidence we have in the product" [17, pg. 6]. For this perspective, testing can be viewed as a mechanism that facilitates the building of trust

in the system being developed. In the context of software development, the trustor is the developer, user, or sponsor of the software system—henceforth referred to as the software owner—and the trustee's role is taken up by the software system itself.

If I was given a guarantee that the system will work exactly as designed, testing would be fruitless. The reason that we give importance to the testing activity is because we want to ensure that the system is indeed working the way we expect it to work. We want to gain that sense of confidence in the software system that is built. Thus, testing may be viewed as a manifestation of the trust that a software owner strives to obtain from the system being built.

Until now, testing has been popularly defined as the method of verification and validation that helps obtain an assurance that the software built meets a certain quality standard. This definition imposes limitations on the interpretation of the nature and purpose of testing. The limitations are imposed because testing itself is not quality assurance, but it informs quality assurance. Additionally, it also provides other assurances including (1) behavioral assurance, that the system will behave as expected in a given situation, and (2) purpose-fitness assurance, that the system will be of utility and value as per its customers' expectations. Quality—identifying defects—although fairly important, may not necessarily be the supreme expectation from the system in all cases. Based on business needs, other factors, such as availability, utility, and ease-of-use, may take priority depending on the software's situation and domain of residence. However, the key idea is that it is important to meet the expectations of the software owner, whatever they may be, and there is a need for a mechanism to ensure these expectations are met. Defining testing as a trust-building approach, provides a broader definition and understanding of the value that testing can render, and may propose a direction to change its second-standard perception (discussed in Section 7.3.1). Consequently, these verification and validation methods become agents in achieving this goal.

Being able to gauge the degree of trust that the developers have in the system they built

is crucial because this degree of trust determines many business and safety decisions. For instance, unless the software owner has sufficient confidence in his software, he may not decide to release it as a ready-to-be-shipped system. Rather, just as Google strategizes its release, the software owner might decide to release the software as a beta version, which means it is still being tested. In other words, a beta release conveys a tacit message from the software owner that means that the developed system is not completely trustworthy at this point. The system might behave different from what is expected or being claimed and it provides a leeway to discount the software owneroff any responsibility resulting from unexpected behavior of the system. For another example, if the software is a part of a safety-critical system, the software owner might just delay the release of the system because they are not confident about the system's behavior under various situations. Boeing's Dreamliner 787 is an apt example of such a situation. Among other reasons that caused a "legacy of delays" in the release of Boeing's Dreamliner 787, software glitches were one common reason. In a recent failure incident in Air India's Dreamliner flight, the airline's representative made the statement that "[o]nce you have lost confidence in the machine and on the software, then you cannot take the risk, so the pilot took the controls in his own hands and landed the plane safely [13]. This "lost confidence in the machine" is an indication of violation of the pilot's trust by the airplane's software system.

In summary, at least in the software-engineering discipline, the meaning of trust extends beyond the idea of the expectations between two individuals (or group of individuals) to individuals and technological artifacts such as software systems.

7.3 Different Cultural Layers of Challenges in GST

In this section, I discuss the three cultural layers that impose unique challenges in accomplishing trust in the GST setting. Section 7.3.1 describes the occupational culture

related challenges, Section 7.3.2 explains the challenges that arise because of the inter-organizational team settings, and Section 7.3.3 presents the cross-national cultural challenges.

7.3.1 Occupational Cultural Layer

Based on the discussion in the previous section, testing may be viewed as a trust-building process, while development may be viewed as a software-building process. Viewing testing as a trust-building process may provide an explanation for the "second-class citizen" culture of software-test engineers that I discussed in Section 2.5.3.

Second class citizens are referred to as groups that are not accorded a fair share of the recognition that they might deserve (i.e., they appear to be systematically discriminated). In the sense of software testing, the share of recognition refers to recognition for the contributions that the software test engineers make to the software-development process. In other words, there is a lack of acknowledgement and recognition in the community of the value-add that the test engineers provide to the software-development process. In fact, evidence from my field-data analysis, as well as anecdotes from practitioners in this domain, buttress the existence of this behavioral trait of lack of acknowledgement and recognition. My participant practitioners explained that testing "was seen as a wasteful exercise. . . it was seen as a mandatory evil" (S2T1.PM1) by some of their clients in the sense that it was imposed by some institution despite being undesirable. "Mandatory evil" indicates that some institution imposes a force to perform software testing, despite the desire to avoid it. Industrial experts claim that testing as an isolated activity, although associated closely with quality, has no inherent value in the industry. [1] Such perceptions appear to be more common among software owners whose primary business is not in the software domain (e.g., retail domain software owners), which still form a significantly large group of software owners. These owners are generally not fully convinced that testing is required. "The question that

[1] Scott Barber presented this idea at his keynote presentation in the "Let's Test 2013" Conference.

they [software owners] ask is why do I need to test. If the application is developed then why do I really need to test? (S3.CMG1)." Thus, the business value of testing is not clearly understood by some stakeholders engaged in the software-development process.

A trust-building system is required when an individual wants to build trust in an entity. What if the individual already has trust in the entity? Then, the trust-building process is a wasted effort because the outcome of the process is already an existing state. It appears that similar dynamics are playing a role in the existing belief system of some members of the software community. These members—some software owners—expect that the software, when built by the developer, will be perfect and defect free. They already trust that the software system will behave as expected. In fact, this misconception is described by some industry experts as people believing "that without any testers, there wouldn't be any errors, [1] " and it is reported to be one of the major conflict zones in testing organizations. Hence, they fail to understand the reason for them to spend more effort, time, and money in ensuring that the system will actually work the way it is expected to work, when they already believe that the system will work the expected way. The software owners might understand that the quality of the system might not be exceptional, but they are ok with a lower quality product because the industry usually appears to value profit over quality. This failure to understand the value of testing perhaps leads to the interpretation of testing as a "wasteful exercise," leading to the formation of the idea that testing is a second-standard job because it doesn't add any meaningful value to the software-development process. Surprisingly, the service-level agreement document of the U.S. Client (S3) Team and Mexican Client (S3) Team stated that it was the responsibility of the vendor teams to "showcase value add and ROI [return on investment]" of the testing, which shows testing advocates' current struggle in the industry.

The perception held is that as long as a high quality system is developed there is little reason to justify not trusting the system that is developed, and thus invest in testing. Thus, the challenge for test engineers is to convince the software owner not to blindly trust the

system that is being built. Asking the software owner not to trust the system that they invested heavily in building indeed becomes a tough sell and this forms one of the important hidden challenges for the testing community.

Interestingly, my study analysis provided evidence that indicate that this conception of "wasteful exercise" is culturally influenced. It was clear in the vendor teams description of their Japanese Client (S3) Team (Section 5.1) that the Japanese team, known for not readily trusting others, did not view testing as a "wasteful exercise." In fact, they preferred to perform exhaustive testing as compared to other clients (e.g., U.S. Client (S3) Team). Being from a culture known for possessing fragile trust characteristics, the Japanese teams relied heavily on this confidence-building system of testing to inform their trust judgements about the system being developed. Consequently, in the Japanese culture, test engineers (commonly referred to as the quality assurance team) that form their "Kensa testing group" possess a higher social status as opposed to the "second-class citizen" status possessed in other cultures. This social status is apparent in the vendor team members' description of experiencing "nightmares" when the "Kensa" team reported a bug. Thus, because trust is culturally influenced, the perception of testing is also culturally influenced.

It appears that the interpretation of testing as a wasteful or a trust-building exercise is a ramification of how the value-add of the testing activities is internalized by the software owner. Although this trend of perceiving testing as a "wasteful exercise" has been changing in the past decade, even in the current day, the service-provider community encounters a different level of challenge in the same space of perceived value-add. "Somebody who has developed [it], should be able to test it. Why do we need independent testing? (S3.CMG1)." This challenge is evident in the service-provider organization's expression of interest in investigating similar issue further. For instance, a client-facing manager participant shared an idea with me, namely that I should investigate whether "there is no value in offshore based testing engagement," which emerged based on the extensive interactions he has had with several clients (S3.CMG2). This hypothesis represents the continued struggle of the

software-service organizations, and the thoughts and common notions carried by many client teams even today.

In the next section, I investigate the potential reasons for the existence of these struggles in vendor organizations such as the ones I studied.

7.3.2 Inter-Organizational Cultural Layer

When viewed as a manifestation of trust, several parameters integrated in the testing activities emerge to be important from the trust perspective. These parameters are particularly crucial in the service setting where the vendor and client organizations are different. The way in which these parameters are handled in these inter-organizational team settings when planning testing activities directly influences the extent to which trust formation occurs and the manner in which trust is maintained. I discuss four such parameters that I identified to be crucial determinants of trust in inter-organizational team settings such as the GST teams that I studied. These parameters, which I discuss in the form of four questions, include the state of testing, amount of testing done, agents performing the testing, and the procedure used to conduct the testing.

With respect to trust, there is one crucial aspect that differentiates the inter-organizational setting, such as the client-vendor team setting that I studied, from the in-house testing practice. This aspect is that in the inter-organizational setting the trustor is still the client or the software owner but now the role of trustee involves two components—the vendor organization and the software system being built. This new dimension of the vendor organization influences many factors of trust dynamics. I discuss how this additional dimension affects the four parameters that I present in this section.

7.3.2.1 State: Whether Testing is Done?

If a user is told that the system they are going to use is not tested, the implicit message being conveyed is that the system might behave in an unexpected way because confidence in the system is relatively low. Thus, whether testing is done or not conveys significant

information about the extent of the level of expectation and degree of trust that the user can set in the system. However, the guarantee that when told testing is done, it is indeed done is a subtle but equally significant element. This guarantee is obvious and hence insignificant in the in-house testing. Nonetheless, in the client-vendor setting this guarantee becomes a concern.

In the context of client-vendor team setting (e.g., GSE), the information of whether testing is done or not has special significance mainly because of the nature of testing. The nature of testing is such that the only outcome of the test execution process is the status of test case—passing or failing. This unique property of having only a status as the final output, instead of an artifact as is the case in software design (e.g., design specifications) and software development (e.g., the code), makes this process vulnerable and thus a candidate for trust violation practice.

When testing is being done remotely, clients frequently doubt whether testing is actually being done. Because of the ease with which a test case status can be manipulated and the lack of a strong binding between the test-case execution process and test-case results, there is usually a lack of trust in terms of the guarantee that testing is done. To establish this guarantee, the GST vendor teams started the practice of providing screenshots of every step in the test-case execution process. Although this screenshot capture process is significantly time consuming and memory intensive, it provides an important value to the GST practice because it mediates trust between the two involved teams (client and vendor) by representing the instituted model of trust.

Additionally, despite the screenshot capture process being widely accepted and implemented in these vendor organizations, it does not necessarily address trust-related concerns of their clients completely. The degree to which this process addresses the trust component depends on whether and to what extent the screenshot capture expectations are explicated in the service-level agreement between the client and the vendor. In Study S3, the service-level agreement document of the U.S. Client (S3) Team and Mexican Client (S3) Team

did not mention anything in relation to the screenshot capture activity. [2] Nonetheless, the vendor organization had committed to providing this additionally-featured service to their client team. This commitment, which is a manifestation of expectation building, played a critical role in the trust-relationship between the two teams. After raising the expectations of the client team, when the vendor team failed to meet this expectation—failed to provide the screenshots— the client's trust was bound to shatter. Despite trust being formed gradually between the client and the vendor, this mere absence of the screenshots when the clients expected them triggered distrust in Mexican Client (S3) Team.

Interestingly, some client organizations' expectations are higher than the vendor teams' publicized services. For instance, in the same example of screenshot capture, vendor manager (S3T2.PM2) from Vendor V3 shared that some of his previous clients had also demanded printouts of the screenshot with the vendor-side associate's signatures on the printed documents to be shipped with the other deliverables. This expectation reflects on the degree of trust these client teams had in their trustees—the vendor teams.

In conclusion, when testing is viewed as a trust building process, test case results become the decision system that affords or hinders the trust in a system. However, in case of the GSE-like setting, an additional important layer of trust emerges—the layer of the support systems, such as screenshot capture—that guarantees the occurrence of the step-by-step execution of the test cases. Moreover, the degree of service expected from the vendor with respect to the screenshot capture system also becomes an additional factor indicating and determining the extent of trust that resides between the two teams.

7.3.2.2 Agent: Who is doing the testing?

Because testing forms the medium for building trust in the system being tested, the agent responsible for this trust-formation process becomes significantly important. A critical agent, involved in ensuring appropriate testing is done of a system, is the test engineer

[2] I do not know the status of the Japanese Client (S3) Team because I did not have access to their service-level agreement document.

himself. The test engineer's extent of understanding of and familiarity with the system determines how well the system will be tested. If a test engineer lacks understanding of the system and its expectations, he might test incorrectly leading to an expected behavior to be tagged as unexpected or vice versa. On the contrary, if a test engineer is quite familiar with the system, he might be highly biased and retain some misconceptions designed into the system that should be tested against in the first place. Consequently, the outcome of the test cases—pass or fail—changes the perception of and trust in the system.

When considering the *agent* parameter of testing, three characteristics of the test engineer appear to be important in the trust-building process: (1) familiarity with the system, (2) proximity to the software owner (particularly, users) of the system, and (3) personal presence in the testing task execution.

It is now a widely accepted view that the developer who builds the system is not the best candidate to perform testing (beyond unit testing) because he is exposed and native to the system. Hence, he would be biased towards the system. Being biased means that the developer understands the system quite thoroughly and hence he knows what will work and what will not work: he knows what to expect from the system. This bias blinds the developer from thinking to test the system beyond its expected behavior. In other words, because the developer possesses deep-level knowledge of the system, he trusts the system too much that it becomes a challenge for him to test the system outside the arena of expected outcomes. In Beizer words, "knowledge... brings efficiency to testing but also blindness to missing functions and strange cases [17, pg.12]. Hence, when the system is tested by the developer or the developing organization, there appears to be a lower level of trust in the system. In fact, the GST vendor organizations appear to be targeting this trust-influencing characteristic of familiarity to promote their business and services. These organizations have been selling the idea that their reduced level of familiarity with the system being built is indeed an advantage for them, enabling them to provide better independent testing service to their clients. The continuously increasing business of independent GST services

may indicate that indeed their reduced level of familiarity is an asset in providing better testing services, which in turn fosters trust in the system.

While it is preferred that the test engineers are less familiar with the system, it is also preferred that the test engineers are in close proximity with the business (e.g., software owners, users). Proximity to the software owner refers to the degree of close association of the test engineer to the business knowledge and environment of the software system being tested. Without an appropriate understanding of the business, it is difficult to understand what to expect from the system; if the test engineer does not understand the expectations from the system well, it is difficult to trust that the system will indeed be tested properly. Thus, proximity to business knowledge influences trust in the testing conducted, which in turn impacts the confidence built in the system.

In the GST setting, which typically involves outsourcing and offshoring, the client and vendor organizations are separated by geographical distance. Consequently, the medium of learning about the client's business lies in strategic practices such as knowledge transfer sessions and client/vendor organizations' visits. Despite these strategic practices, it becomes challenging (1) for the vendor teams to acquire detailed understanding of the client's business and (2) for the client team to believe that the vendor teams thoroughly understood the business needs to test against. For instance, in the Study S3, the Mexican Client (S3) Team struggled with gaining the confidence that the offshore vendor team understood the business correctly. Hence, the client's (U.S. Client (S3) Team and Mexican Client (S3) Team) parent organization wanted to retain the accountability of testing at the client side, thus practicing the strategy of performing the first cycle of testing at both the client and vendor side. However, interestingly, the Mexican Client (S3) Team, being a relatively new engagement as compared to the U.S. Client (S3) Team, was still not ready to trust the testing that the vendor team was doing and hence it had continued to perform the testing activities beyond the first cycle. This action of the Mexican Client (S3) Team demonstrates that they were not able to trust the system because they lacked confidence in their agent—the vendor

team—that was performing the testing. In addition, the vendor team's novice members (i.e., fresh college graduates) asked basic questions that showed their lack of understanding of the business. As a result, this action further reduced the Mexican Client (S3) Team's trust in the vendor team's capabilities to perform testing appropriately.

Thus, it is evident that the test engineer is a critical agent in determining trust dynamics in the GST setting. This critical role of the test engineer is amplified in the automation testing practice. In Study S2, the U.K. Client (S2) Team was new to automation testing practice, and hence they were doubtful that an automation script will test the system properly. The idea, which the client team had a challenge in believing, was that an automation software could be used instead of a real test engineer to test the system. The absence of the test engineer for testing the system created suspicion of the manner in which the system would be tested, thus leading them to a lack of confidence that the system would be tested correctly. In conclusion, the agent's presence, proximity, and familiarity play a complex but important role in forming the trust in the system being tested.

7.3.2.3 Amount: How much testing is done?

The thoroughness and extent of exhaustive testing may be interpreted as the extent of trust one has in the software system being built. If a software owner has higher degree of confidence in the built system, he may not seek to test the system thoroughly and vice versa. This confidence level builds on factors such as quality and these factors are based on perceptions that vary across organizations and businesses (discussed in Section 5.1.2.2). It is practically impossible to perform complete testing of any system and hence the amount of testing that may be performed on a software system is an organizational determinant.

The amount of testing includes the number of test cases designed, planned, and executed in a test cycle and it reflects the extent of trust built in the system. The greater the number of passing test cases, the greater the confidence in the system, and vice versa. Moreover, software complexity influences the perceived trust in the software system as

well. More complexity results in less trust that the software will perform as expected and hence more testing is demanded in such cases.

In the cross-organizational setting, the amount of testing contributes to a new challenge with respect to trust formation. Because two organizations are now involved in this process, the perception of what is a sufficient amount of testing—how much is "good enough"—may differ across organizations. In fact, my vendor participants from Study S2 reported such a difference. The vendor participants thought that the clients' decision related to the testing amount were incorrectly estimated. Despite the vendor teams performing testing to meet their client's expectations, the activity lead to the unproductive productivity experience that I discussed in Section 6.1. Thus, in cross-organizational setting, deciding which amount is the right amount of testing becomes a challenge because different organizations perceive different amounts of testing as sufficient for forming the trust in the system.

Interestingly, based on my field studies' analysis, it was evident that deciding how much testing is enough was predominantly done by the client organization. The vendor organization participated in providing suggestions but they were at the disadvantage of not being able to contribute equally to the testing scope's decision-making process. This disadvantage was a result of their reduced awareness of the client's business to the extent that was required to make such decisions [94]. Nonetheless, the vendor organizations were now making effort to better understand the businesses and become clients' "trusted advisors." This new strategy of the vendor organization to become client organization's "trusted advisor" indicate the vendor organization's difficulty in providing an accurate estimate of the amount of testing that the client believes is enough to form trust in the software being tested.

In addition, as discussed earlier, because of the additional dimension of the vendor organization as the trustee, there is an additional factor that acts as a trust determinant with respect to this *amount* parameter. This factor is the number of defects found in the system. The vendor organizations use defect estimates from previous projects to gain the

confidence in the capabilities of their potential future clients. For instance, the corporate marketing group member of Vendor V3 (S3.CMG1) stated that if they can assure their clients that they "will find 1% of the defects in [their] production system going forward, then that gives [them] a kind of confidence in the [Vendor's capabilities]." Thus, showing evidence suggesting that the vendor organizations are capable of finding most of the defects before production illustrates that they are indeed good at testing (i.e., building the trust in the software being tested).

7.3.2.4 Procedure: How are test cases steps and environment replicated?

Testing may be viewed as a process that involves replicating and reproducing potential real-world situations and checking whether the software system will behave as expected in such situations. Two important ingredients of this systematic replication and checking process include (1) the *test-case steps* that are used to replicate the interaction of the software system and (2) the *environment* with which the software system interacts. Incorrect replication of either the steps or environment might result in false output of the test case execution, which would falsely influence the software owner's confidence in the system. Hence, the closer these steps and environment are to the real-world, the more value is contributed by testing.

However, cross-organizational setting impose significant challenges to correct replication of test-case steps and environment. Some challenges emerge because it is difficult to create identical infrastructures at the vendor organization's location because of concerns such as cost and privacy. In fact, my study analysis reinforces this finding. The analysis revealed that the Mexican Vendor (S3T2) Team did not have access to the required infrastructure and test data for testing. As a result, they tried to implement workarounds to proceed with their testing activities and meet the scheduled deadlines. However, the workarounds did not replicate the real scenarios closely. Unfortunately, this failure in replication was misinterpreted by the Mexican Client (S3) Team as lack of capabilities of the vendor team

to perform correct testing, thus losing trust in the vendor team. Thus, although the failure was in underestimating the challenge in replicating the environment, it was interpreted as the failure of the capabilities of the vendor team.

A good understanding of test-cases steps to be executed is also a crucial factor for forming the trust in the system being tested. In my study, when the "fresh graduates" test engineers asked the client some clarification questions with respect to the steps involved in the test case execution, the Mexican Client (S3) Team interpreted that as the team's lack of understanding of the software system and again doubted the capabilities of the vendor test engineers to perform testing correctly. This incident indicates that understanding the manner in which test-case steps are executed also plays a critical role in the trust-formation process.

The fact that the Mexican Client (S3) Team continued to test the software system beyond the first cycle indicates that they still lacked confidence in their software and doubted the capabilities of the vendor team members. Thus, to build trust in their software, they chose to test the system in addition to the vendor team's testing effort.

7.3.3 Cross-National Cultural Layer

Al-Ani states that "culture implies a shared understanding that is used as a baseline from which expectations are set" [8], where her reference to culture hints at the national/regional cultural dimension. Thus, the foundation of trust, which is manifested in the form of expectations, is significantly influenced by the national cultural systems. In the GST setting, because the team configurations commonly consist of members from different nations, different members possess their own national cultural foundations that shape their trust perceptions.

In reference to the trust and testing discussion presented in this chapter, the cross-national cultural dimension emerged as a vital dimension inevitably influencing trust dynamics in the GST settings. My analysis suggests that this cultural system has a direct,

as well as indirect, influence on trust dynamics. The dynamics are directly influenced by impacting the client-vendor relationship in a GST setting and indirectly influenced by impacting the occupational and inter-organizational level challenges discussed earlier. In this section, I present the analysis providing illustrations of these direct and indirect influences of the cross-national cultural dimension on the trust relationship in the GST practice.

7.3.3.1 GST Business Acquisitions in Certain Geographies

At my participant organizations, most of the teams I studied had cross-national team configurations. My vendor participants had to face different degrees of struggles to gain, regain, or maintain the trust of their clients based on their culture and geographies. In addition, an interesting aspect was that the degree of effort invested in handling their credibility was not associated with the length of the client-vendor partnership. In other words, a longer duration of the client-vendor relationship did not necessarily mean that the vendor team would have to put in less effort to handle the clients' trust, which is counter intuitive. In fact, my findings illustrated that for the Mexican Vendor (S3T2) Team—1.5 years of partnership—and the Japanese Vendor (S3T3) Team—10 years of partnership—accomplishing trust of their respective clients was a challenge, whereas the U.S. Client (S3) Team—3 years of partnership—had to invest comparatively less effort to accomplish the trust of their clients.

The vendor teams had to face challenges in terms of gaining the clients' trust, particularly clients from geographies such as Japan. These geographies were big business markets for the vendor organizations, but they constantly faced hurdles in getting projects from them. At one of the talks addressed by a board-level member of Vendor V3 organization, he expressed that "Japan was the third largest country economy wise and second largest country from the Information Technology perspective." Despite Japan being a potential business market, the organization has been battling to build trusted relationships with Japanese companies. This is evident from data on the geographic distribution of Vendor V3's clients that used their independent testing service. With respect to the testing service, the data reported

that 63% of the Vendor V3's clients were from the U.S. but only 1% of the clients were from Japan. These statistics corroborated the cultural training team trainers' information, who reported that they frequently received trust-related complaints from teams working for Japanese clients.

The reason for these trust challenges can be traced back to the cultural characteristics of the Japanese as being uncertainty-avoiding members in a closely-connected society (refer to Section 5.1.2.1). Because of these characteristics, the Japanese people may be skeptical of trusting the testing of members outside their group and hence they may have been reluctant to outsource their testing services to offshore organizations such as Vendor V3.

7.3.3.2 Language as a Catalyst/Hurdle for Trust Formation

Language reflects culture because culture is said to be transmitted through language. Team members speaking the same language have closer connections with each other because they better understand the tacit meanings and unspoken expectations that exist in conversations. Consequently, trust would be higher within groups that share the same language.

My participants reported their experiences working with clients that were not from English-speaking nations. Because my participants' first language was not English (they were all Indians with different regional mother tongues and a decent degree of proficiency in English), when they interacted with clients whose first language was also not English, they faced additional challenges in gaining the client's trust. Such a struggle is evident in the experience narrated by one project manager (S3T2.PM4) from Study S3 who was interacting with a German potential client to start a new testing-service engagement:
(S3T2.PM4)

I'm coming back to the language issue... that also has been a major problem... If you can't connect with them, if you don't speak the language that they understand then it becomes very difficult. I was working on a proposal for a German client... It was very difficult, the kind of detailing that we had to do for the proposal going off to the German clients was far more than what we would have done otherwise. We had to tell them almost everything to the last level of detail that we can so the proposal itself would run into hundreds of pages. So that is the kind of detail that they look at to decide on whether they really want to give it [the project] to you or not and whether you understand their [business] problem and whether what you are suggesting to them is relevant or not. So the factor of geography definitely comes in. (Quote 36)

This demand for high degree of detail indicates that the German client—the trustor—was apprehensive about the vendor team—the trustee—and they were seeking confidence that the vendor team understood their expectations. The non-overlapping languages caused a barrier to their communication of these expectations. As a rescue strategy, the German team appeared to rely of the commonly shared language—English—and demanded documentation explicitly stating every expectation "to the last level of detail." Making these expectations explicit was the method they adopted to initiate the trust formation process.

My participant vendor organizations appeared to be aware of this role of the language in trust formation. This awareness might be the reason that the vendor teams were investing effort to learn their client's language. For instance, some members of the Japanese Vendor (S3T3) Team had undergone Japanese-language training. For another instance, there were Spanish-speaking classes being conducted on the campus while I was conducting the study at Vendor V3. Although efficiency in communication was one reason for learning the language, trust building by demonstrating vendor organization's willingness to learn client's culture was an additional reason to undergo language training.

Thus, the language manifested itself as a vital agent transforming the trust relationship in the GST setting.

7.3.3.3 Cross-National Cultural Manifestation in Inter-Organizational and Occupational System of Testing

The study analysis discussed in Section 7.3.2 suggests that the testing parameters such as amount and procedure differed across organizations. Although these characteristics differed across organizations, it did not necessarily stay the same within an organization. Comparative analysis of the two teams—U.S. Client (S3) Team and Mexican Client (S3) Team—within the same parent organization indicated that the two teams' expectations with respect to parameters such as the amount of testing differed. This difference was evident in the surprised behavior that the U.S. client member (S3.CO1) demonstrated when she realized that the Mexican Client (S3) Team's expectations with respect to testing plan differed from their plan, resulting in a testing practice that differed from the U.S. Client (S3) Team. The difference was that the Mexican Client (S3) Team was performing more testing at the client site in cycle three than the U.S. Client (S3) Team counterparts.

Nonetheless, the cultural characteristics of the clients from the same geography demonstrated similarities. For instance, participants from Vendor V3, serving different roles (e.g., board members, project manager, and test lead), who had the opportunity to interact with Japanese clients mentioned that most Japanese clients' standard of quality were extremely high and, thus, it was difficult to gain their trust in terms of testing and quality. Such characteristics indicate that the national cultural dimensions dominate the organizational cultural practices, illustrating the indirect impact of cross-national cultures on trust development [27].

From the occupational perspective, the perception of testing within the different teams of the same vendor organization appeared to differ. The vendor teams working with Japanese Client (S3) Team mentioned that based on their interactions with the Japanese members, they had started realizing the importance of conducting exhaustive testing as practiced by their Japanese client teams. In fact, they described situations where they were able to learn

new strategies for performing efficient and exhaustive testing. Also, based on their description, it was evident that the Japanese Client (S3) Team did not view testing as "second standard." In fact, as discussed earlier, the "Kensa" testing team was accorded higher social status. However, at my participant vendor organizations in India, the experiences reported were different. In the description of their experiences, it was evident that test engineers were not perceived with high regards: "however, in India, if you are a tester for such a long time, then you are perceived to be not capable to do other roles [e.g., managing or leading a team] so you are a tester (S2T3.PM1)." (refer to the discussion in Section 2.5.3).

Thus, the national cultural perspective influenced both the inter-organizational and the occupational perspectives that impacted the trust accomplishment dynamics in the GST setting.

7.4 Discussion: Reasons for Violation of Trust in GST

As discussed in this section, trust accomplishment in the GST practice encounters challenges across different layers—national, organizational, and occupational. Because of the global setting, the trust accomplishment key players change as well. As discussed earlier, in this global setting the trustor is still the client team but the trustee is composed of the software system as well as the vendor team. Because of the influence of the intricately intertwined cultural layers as well as the presence of this distinct set of key players, trust accomplishment in the context of testing becomes a complex system.

In this new global configuration, frequently the trust focus appears to move away from the software system to the agent who ensures trust enforcement. The concentration then shifts from monitoring the behavior of the system and to monitoring the actions of the test engineers performing the testing. This explains the expectations of some clients (e.g., Mexican Client (S3) Team) that they receive daily and weekly status reports of the progress and status of the testing from their vendor teams. Such monitoring and cross-checking activities have been reported to decrease effectiveness in distributed teams [8]. However,

trust violations occur mainly when the trustee fails to meet the expectations of the trustor. Several reasons provide an explanation for the failure to meet the clients' expectations in the GST setting. In this section, I present a discussion of some factors, which I identified based on my study analysis, that provide potential explanations for vendor team's actions that led to trust violations in the Study S3. These explanations can be a starting point to investigate similar other trust violations that are reported to be occurring on frequent basis in GSE settings similar the ones I studied.

7.4.1 Mismatched Thresholds

One vital but tacit characteristic of this complex trust system is the *determination of the minimum threshold of expectation* that must be met to accomplish trust. This threshold of expectations has an unstated presence and it manifests itself in various forms in the GST context. For instance, the idea of identifying a "good enough" amount of testing is one such manifestation of this threshold of expectations, where determining this "good enough" amount represents the determination of the minimum threshold of expectation with respect to the extent of effort required to build the desired trust in the software being tested. Because exhaustive testing is practically impossible, this threshold becomes a vital determinant in the testing domain.

Different approaches are adopted in industry to tacitly identify such thresholds that can help build the desired trust in the system. The process of test planning that involves determination of decisions such as what will be tested, who will test it, and how much testing is sufficient becomes the medium for identification of the threshold. Although detail studies of the client teams is required to obtain a deeper understanding of the adopted approaches, it was evident from my study analysis that this threshold varied among the clients as well as vendors, and were further governed by the three cultural layers—national, organizational (and business), and occupational. For instance, the negotiations between the client and the vendor about different parameters within the testing paradigm, such as whether testing is

required and how much testing is required, was indeed an external manifestation of this threshold-determining process.

Furthermore, different internalizations of the testing thresholds by the client and the vendor teams appeared to have led to the unproductivity productivity experience, which I discussed earlier (Section 6.1). The test engineer (S2T3.TM1) in Study S2 seemed to have experienced unproductive productivity because he and his client team member had different internalizations of the expected amount of testing thresholds. His threshold was lower than that of the client member—"So for 5-6 [test] cases, I am going to write 100 some test cases ... this number will look good but you know it's not logical work... the goal will be the same but effort and visibility will be more in 100 test cases" (Section 6.1, Quote 26). Despite differences in the internalized thresholds of expectations with respect to the software testing process, S2T3.TM1 performed the testing under the governance of the Agreement Cultural Model to meet the expectations of his client member. This illustration depicts the existence of two levels of thresholds of expectations—(1) between the client and the vendor and (2) between the client and system, and the vendor and system. Although the testing thresholds between the client and system, and the vendor and system were different, this difference did not surface because of the Agreement Cultural Model's internalization, under whose influence the vendor member met the client's threshold of expectations.

In conclusion, different cultural layers were ultimately altering the thresholds of expectations for the client and vendor team. The extent of overlaps between the client's and the vendor's thresholds of expectations governed the inception of experiences such as unproductive productivity and dominated the extent of effort the vendor required to invest to meet the trust related expectations of their clients. Hence, it is crucial to identify and unpack the characteristics that form the thresholds and, more importantly, it is essential to make these tacit thresholds explicit to ensure trust dynamics are not violated.

7.4.2 Imbalanced Distribution of Importance to Trust-Related Parameters

As discussed in Section 7.3.2, in the inter-organizational setting, many parameters surface as important trust-accomplishment factors. Although the current data lacks sufficient information to make any conclusions about the ranking of importance of these parameters, it at least provided evidence suggesting that these parameters were all important. However, the vendor teams' actions suggested an implicit assignment of priorities or ignorance to the importance of some trust-related parameters. Several instances illustrate this imbalanced distribution of importance. I discuss some vignettes from the Study S3 below.

First, the agent of testing is an important trust determinant. However, the Indian vendor teams' strategies of including "fresh graduates" on client projects (even new engagements such as the Mexican Client (S3) Team project) demonstrated their ignorance of the *agent* parameter and its significance in the trust-accomplishment process. Upon learning that inexperienced members were working on the project, the fragile trust of the Mexican Client (S3) Team was ruptured, leading to client decisions and behaviors that surprised the vendor teams because they were perhaps ignorant of the significance of this parameter in the trust accomplishment process.

Second, trust and distrust are said to be distinct and can co-exists. This distinctness was evident at least in the Mexican project because elements of both were present in the Mexican team simultaneously. The Mexican Client (S3) Team trusted the capabilities of the Indian vendor organization, based on the recommendations received from the U.S. Client (S3) Team members. This trust was evident in the fact that the Indian vendor team got the contract to perform the independent testing for the Mexican Client (S3) Team project. In the absence of this initial trust, they would not have outsourced the project to Vendor V3 in the first place. Nonetheless, at the same time, they were skeptical of Mexican Vendor (S3T2) Team's capabilities. Hence, they were the ones to initiate the practice wherein the client team would perform the first round of testing always. Also, compared to the amount of testing outsourced by the U.S. Client (S3) Team, the Mexican Client (S3) Team

was very wary about outsourcing testing activities, which concerned the Mexican Vendor (S3T2) Team. Again, this concern appeared to have been aroused because of the ignorance of the vendor teams that trust and distrust can co-exists.

Third, the vendor team's internalization of the meaning of trust appeared to have been restricted to the *amount* parameter (Section 7.3.2). Hence, they concentrated heavily on meeting the threshold of expectations with respect to the amount of testing expected weekly that was specified in the test plans. They put additional effort to ensure that they did not have any "red status" indicating that they were not able to meet a particular planned deadline. However, in this process they appeared ignorant about the other parameter of *state* and hence neglected such parameters when they were under pressure to meet deadlines. Consciously taking the decision of missing the screenshot capture process is an illustration of such a negligence. There were two levels of issues that led to trust violation here: the vendor teams (1) were ignorant of the role of state parameter in trust accomplishment and (2) had internalized an implicit levels of priorities with respect to the identified trust parameters. These issues explain why the Mexican Vendor (S3T2) Team were surprised and in disbelief when the "losing credibility" issue emerged and were left wondering why the clients were behaving the way they were even when "we [they] have delivered" (S3T2.PM2).

7.4.3 Lack of Explication of Expectations

The study analysis revealed that many expectations across the teams were implicit, leaving opportunities for teams to interpret the expectations differently.Because sometimes teams internalized different interpretations of the expectations, a gap appeared to have existed between the client teams' expectations and the vendor teams' understanding of clients expectations. This gap eventually became the reason for trust violation in some cases.

For instance, consider the screenshot example. The service-level agreement did not explicitly mention the requirement that they expected the vendor team to provide the screenshots. However, an implicit expectation was built into the client teams. I speculate this

expectation was held by the vendor team when they advertised the screenshot capture practice as a featured service that they provide along with the testing service. Because this practice was not internalized as a formal commitment, the vendor teams appeared to view it as an add-on optional feature, which took lower priority during deadline situations as compared to the higher priority tasks of delivering the test-execution and its results. However, this was not the case at the client end. For the clients, it appeared that the screenshots had taken up the role of being the instituted model of trust. In addition, the screenshot capture process was implicitly addressing the *state* parameter related to inter-organizational trust-accomplishment challenge discussed earlier (in Section 7.3.2). This mismatch in the internalization of expectations led to the trust violation when the vendor team failed to provide the screenshots. This mismatch also provides possible explanations for the vendor team's astonishment at the client team's behavior of performing testing during later cycles.

Similar to the expectations with respect to screenshots, there were other expectations that were implicit: the official contractual documents (e.g., service-level agreements) did not mention anything with respect to the experience of the vendor team members at offshore. Thus, these unstated expectations risked the trust relationship between the client and vendor teams.

As recommended by other GSE researchers such as Al-Ani et. al. who have investigated trust dynamics in similar settings, one solution for avoiding such trust violations is to "establish practices in which proposals, conclusions and task assignments are explicitly articulated and documented at each location" [8]. The study analysis indicates that such violations can be avoided when the teams invest effort in reiterating their expectations and confirming their interpretations of commitments—by making their expectations explicit.

7.5 Summary

Despite technological advances and support, temporal and geographic differences create several challenges, many of which manifest as tacit cultural problems. Addressing these

cultural problems, whose presence is often unknown or unacknowledged in the industry, become essential for successful execution of GSE projects. The cultural system manifests itself in various forms and across different layers and configurations, and intertwines closely with the technical system of software engineering, making the problem of investigating cultural influence in GSE practice a complex problem.

The cultural models approach facilitated me in performing this investigation. In the previous chapter (Chapter 6), I discussed the three cultural models—Trust, Agreement, and Flexibility—that I discovered, which had a significant influence on how the GST practice unfolded at my participant vendor organizations. In this chapter, I further investigated the Trust Cultural Model. I presented an illustration of the complex cultural system of trust and testing by unpacking the different cultural layers of challenges encountered in the trust-accomplishment process. Particularly in the context of GST, I identified three factors that provide explanation for trust violation in the GST practice that I studied: (1) ignorance of threshold of expectations, (2) negligence in imbalanced distribution of importance to trust-related parameters, and (3) lack of explicitness in stating expectations. Thus, the identification of such cultural models, the understanding of the interactions among these cultural models, and the discovery of the parameters that influence these cultural models are crucial steps towards comprehending the GSE practice and the embedded cultural challenges.

CHAPTER VIII

CONCLUSION AND IMPLICATIONS FOR FUTURE WORK

It is known that there is a disparity between the theory and the practice of software engineering [106]. Prior explains this disparity in her dissertation:

> "Striving for a grand theory of SE [Software Engineering], which includes the ideal software process that disembodies software development work, and examining human behaviour by treating it as equivalent to machine behaviour, are both strands running through software engineering research. The effect of these underlying precepts mean that the technologies and techniques that we design and produce to support software development work may fail to do so effectively, for reasons that have fallen outside of our field's understanding of software practice. [106, p.38]."

The primary reason for studying a particular practice is to understand and recognize that there is a powerful relationship between how a practice is conducted in the real-world context and the design of effective and efficient solutions to support and enhance that practice [106]. However, this relationship is often overlooked or neglected, which leads to the creation of innovative solutions addressing a wrong set of problems. Evidence from literature suggests that technical and social systems are interconnected forming a complex web of socio-technical systems of software-engineering (e.g., [54, 106]).

The ethnographic investigation that I conducted in this research and the existing literature (including literature outside software-engineering discipline) work supports the idea of extending socio-technical software-engineering systems to include the culture component, which plays a crucial role in the global software engineering (GSE) practice. Although this is not the first research attempt at demonstrating the role of culture in the GSE practice, to the best of my knowledge, it is the first systematic, ethnographically-informed, approach towards illustrating cultural milieu in which the GSE practice is realized. My research findings suggests that cultural systems strongly intertwine with the socio-technical systems

to form the complex practice of GSE. Disregarding any one of these systems will paint a distorted picture of the reality of this practice and hamper the goals of improving and enhancing the GSE practice.

The previous chapters, Chapters 4, 5, and 6, presented an illustration of how the technical and cultural systems were interwoven. In this Chapter, I reflect on this findings (Section 8.1), summarize my contributions (Section 8.2), and provide implications, for future research (Section 8.3.1) and practice (Section 8.3.2).

8.1 Reflections

"Culture surprises" us with the myriad ways in which it influences a particular practice being investigated [102]. My findings show how cultural aspects had a subtle but significant impact on the global software-testing practice that is much greater than what was originally envisioned. I present several insights that illustrate how cultural aspects were conveniently ignored or disregarded as being insignificant. Moreover, these insights demonstrated the consequences of ignoring cultural facets and their interplays. I recapitulate some insights below:

- Enacting based on agreement cultural model disrupted the trust cultural model
- Unproductive productivity experience manifested when technical activities were disregarded based on a lack of understanding of their cultural implications
- Overuse of flexibility model resulted in a permanent damage to the offshore teams by raising the clients' work-outcome expectations because they were now expected to work for late hours

In this section, I reflect on two elements of ideas: (1) the experience element (Section 8.1.1) and (2) the cultural-sensitization element (Section 8.1.2). Both these ideas emerged from of my study insights. I concentrate on these two elements because, based on my understanding, these two areas have not yet received the attention they deserve with respect to the GST practice.

8.1.1 The Experience Element

Software engineering is considered to be a human-centric activity where people come together to build a software. The process of performing these activities includes interaction among various involved entities to generate certain kinds of experiences. Research fields such as Human-Computer Interaction give significant emphasis to the experiences that are generated when a user interacts with a technology. These experiences not only inform the practice, but also help improve it. However, GSE research seems to be lacking at giving similar importance to the experiences generated while performing the GSE activities in situ. Here, I illustrate three different instances where the experience element was less focussed.

Unproductive Productivity Experience. Through my research, I found that the cultural facets (Chapter 6) generated experiences such as unproductive productivity among the test engineers. For instance, I describe the screenshot capture example and its association with the Trust Cultural Model in Chapter 6. Although the screenshots were important as they were the external representations of cultural elements of trust, the process of capturing screenshot on every step and uploading them was indeed a time-consuming process, which brought in the unproductivity component of the experience—the bad experience component. Nonetheless, this was considered to be a "normal" practice as "majority of the customers [clients] ask for it [the screenshots]" (participant S3T2.PM2).

While the software-engineering research focuses on improving the efficiency by designing solutions such as novel tools and techniques, effort should also be invested in designing solutions for eliminating the bad experience components. Such screenshot procedure took up significant amount of test engineer's time—based on the collected data, I estimate every test engineer spents a minimum of 17.5 minutes each day in the screenshot related activities. Thus, in a team of 10 test engineers, approximately 3 hours of wastage results everyday per team. Automating such screenshot-capture processes can definitely improve the efficiency of the current global software-testing practice while still preserving the cultural value that it adds to the GST system of practice. Moreover, the solution will still retain

the good experience component that contributes to the trust-building process.

Manual and Automation Testing. For instance, in In Section 4.1.2, I discuss the manual versus automation testing experience. The section describes how manual testing (1) is stilled preferred in the first round of testing and (2) facilitates the experience of training. To the best of my knowledge manual testing is perceived as an inefficient approach for software testing in software-engineering research and academic disciplines, which are more inclined towards the adoption of automation testing techniques. While this inclination is intuitively reasonable, the experience that manual testing brings into the setting cannot and should not be ignored because this experience is facilitating the generation of the value and knowledge that is contributing to the efficiency effort in a uniquely different way—by training the team members in using the system. Thus, research needs to direct attention to designing solutions that retain these experiences while providing efficient solutions because if the efficiency problem is solved but the designed solution removes a valued experience, the outcome is the creation of a bad experience.

Team Structures. The discussion in Section 6.5.1 Mexican (S3T2) Onsite Coordinator Team internalized different cultural models from the Mexican Vendor (S3T2) Team. In Section 4.2, the team structure analysis suggests that the teams with "local hires" as their coordinators (e.g., the Mexican team in Study S3 and U.K. team in Study S2) experienced additional struggles while rendering their GST duties. The study insights suggest that cultural systems influence the team structures, that in turn dominate the experiences produced in the practice. Hence, such experiences, manifesting from different team structures, demand attention. In conclusion, it is crucial to (1) identify the correct experiences and (2) address those component to improve the overall experience.

8.1.2 The Cultural Sensitization Element

Large organizations such as the ones I studied are sensitized about culture's impact and hence they do have infrastructures in place for conducting cultural training. Figure 16

shows one such cultural training room at the Vendor V3 organization. As mentioned earlier (Section 3), I had the opportunity to interact with some members from the cultural training group at Vendor V3 organization.

The manner of dispensing such cultural training sessions reflected on an interesting organizational thought. Any member who would soon be visiting the client side (e.g., becoming an onsite coordinator) was mandated to take this training. Other than this proactive step, the organization appeared not to take any other proactive measures to address the cultural elements. Nonetheless, such trainings gained value when client-vendor relationships were in jeopardy and the teams were not able to reconcile. At such times, these training teams were called in to play the role of mediators whose primary job then would translate to bringing in harmony among the teams by settling the differences. Conversations with the trainers revealed that the being reactive instead of proactive many a times have resulted in failed client relations, which could have been easily avoided.

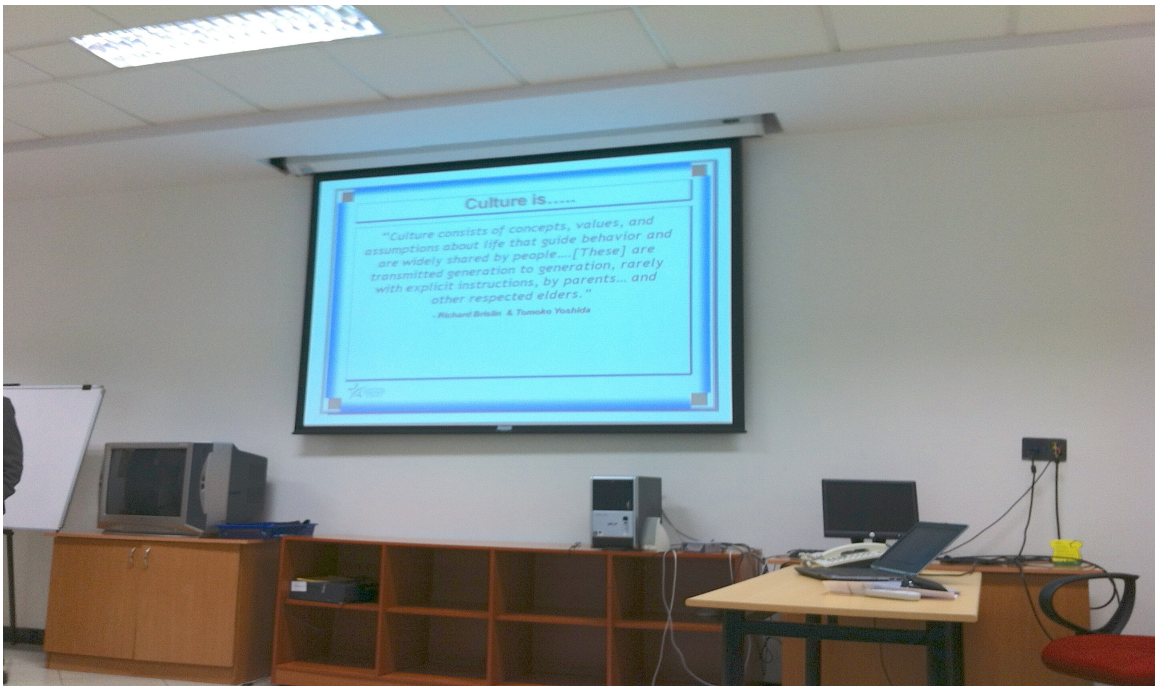


Figure 16: Cultural Training Session Room

The training groups invested tremendous effort in bringing in examples from the field

to explain the cultural influences. Nonetheless, they tend to extract examples that fit the dimensional frameworks such as Hofstede's dimensions. In fact, the training teams consistently used Hofstede's dimensions for their training material to illustrate the cultural differences to the audience, majority of whom were software engineers. Although Hofstede-like dimensional approaches may be great starting points for such cultural-sensitivity discussions, such dimensional approaches have serious limitations as in Section 2.3. In addition, the use of such static dimensions for these training programs resulted in the trainees experiencing disconnection from the contents because the contents may not be addressing the real issues that they would have been encountering consciously or unconsciously. Consequently, such training programs were typically not taken seriously but instead were viewed as an imposed mandate by the organizations. Surprisingly, one trainer (S3.CT2) was mindful of the limitations of such dimensional approaches and she expressed her concern about the outdatedness of work by Hofstede. She shared her thought with me:

(S3.CT2)

. . . I heard it and I thought how dated are they talking. . . I have not done mine looking around [research] to see if there is more updated stuff [literature] or it is only like Hofstede is famous so everybody goes back to Hofstede ['s research] for preparation when they have this kind of [cultural training] session [to conduct]. I don't know if there is any other work available that is more recent but not that famous. Honestly, I think . . . they need to read some better stuff. . . . if there is something better stuff out there then its ok. . . if not, we need to see if we can do something about that. (Quote 37)

Moreover, the training material mainly covered the visible aspects of the culture. Typical topics that were covered in such trainings were client's geography (map, flag, language, climate, time zones), spoken language, currency, public holidays and festivals, greeting norms, social and business etiquettes (e.g., dressing, seating, and meeting arrangements), and preferred verbal and non-verbal communication protocols. These aspects may be viewed as the visible aspects of the cultural iceberg.[1] However, my research corroborates

[1]Cultural iceberg is a metaphorical concept used for describing culture as an iceberg, which has some visible cultural aspects (the part of the iceberg above the sea) but many invisible aspects (the part submerged below the water.)

that understanding only the visible cultural aspect are not enough. The findings presented here provide evidence that the invisible—cognitive—aspects of culture also play a significant role. Thus, there is a need to expand the scope of such training programs to not only cover these visible aspects but also include the cognitive or cultural models "in the mind" by adopting approaches such as cultural models approach.

Another interesting element of these training material was the tone in which the message was being conveyed. I found that the training material had a conventionally prescriptive tone—"Don't do…" and "Should not do…." For instance, the training material used for training members, who would be visiting Japan to work with Japanese clients, included statements such as "Don't break the silence if your Japanese counterpart isn't speaking." It was intriguing to find such prescriptive tones frequently in the training material and presentations that, I believe, typically lead to stereotypes and misinterpretations because the prescription is being provided outside the context of the situation. What if the Japanese counterpart was not speaking because he was indeed waiting for the vendor member to answer? I view this argument to be similar to the winking exemplar narrated in the "Thick Description: Towards an Interpretive Theory of Culture" [51]. Thus, such instances emphasize the need for designing new programs that embed the cultural messages better in the real context.

Cultural sensitivity is expected from both teams—client and vendor—involved in the business. However, my study insights suggest that if the clients are culturally sensitive (e.g., U.S. Client (S3) Team was sensitive), it would significantly help build and maintain relationships and increase the business's success rate because they are the decision makers(I discussed this idea in Section 6.5.3). However, my cultural-training team mentioned that they were conducting few client-side cultural sensitivity sessions—the team was aware of only three client-side training sessions that they had conducted as compared to the hundreds of sessions conducted with the vendor team members. I believe that the

reasons for such fewer client-side trainings are (1) hesitation to ask client teams to undergo cultural-sensitivity sessions or (2) ignorance to the knowledge that such sensitivity trainings contribute significantly to the project's success rate.

These cultural trainings all focused on teaching the software-engineers about the client-side culture. However, what appeared to be missing was making the vendor teams aware of their own cultural models. The lack of such awareness have resulted in serious consequences as it was evident in the escalation sitaution. The project management level manager had informed me that the Mexican Vendor (S3T2) Team and the Mexican (S3T2) Onsite Coordinator Team attended cultural training sessions. However, the escalation still happened. This example indicates that the training need to indicates that the trainings need to focus also on generating awareness about the vendor teams cultural practices.

8.2 Contributions

This section discusses the contributions of this research, the key insights that emerged from the study, and the feedback received from the participants.

8.2.1 Merit of the Research

This research has made several important contributions:

1. A framework that facilitates conducting culture-based studies in the global software-engineering domain
2. A comprehensive description of the cultural models that are embodied in the global software-testing practice
3. A detailed report of the investigation of the complex cultural system of trust and testing, and identification of the different layers of challenges ensued during the trust accomplishment process.
4. A comparison of the cultural models to better understand the supporting and clashing cultural facets in the global software-testing practice

8.2.2 Key Study Insights

The key insights that emerged from this research study are as described below.

1. Current GSE research has lacked systematic investigation of the vendor's cultural system to understand the reasons behind vendor's behaviors and their practice . My study is the first study that concentrated on the vendor side's cultural understanding to investigate the deeper meaning that drove their cultural behaviors and provide interpretations explaining those behaviors. For this reason, my research's scope was to mainly study the cultural practice of the vendor teams and uncover the cultural models embodied in the vendor-side practice.
2. Cultural models framework helped identify the hidden models such as Flexibility Model that are playing a critical role in the GST practice. Such models have not been discovered in the past research on cultural studies, which indicate that the traditional dimensional approaches failed to capture such cultural systems that reside in the GST practice. Moreover, the framework approach facilitated more fluidity than the rigid, static dimensional approach by providing the means to uncover specific models (flexibility, GSD business model) as well as generic (foundational) models (e.g., trust).
3. Adoption of the cultural models approach facilitated the discovery of testing as an embodiment of the trust accomplishment process. This interpretation of testing has introduced a new perspective to understand some observed behaviors of the GST practice. In addition, it has also helped unpack the different layers of challenges that emerge because of the intertwined interplay of various cultural systems including occupational, inter-organizational, and cross-national cultures.
4. The existing GSE research has only highlighted the general aspects of cultural influence (e.g., "always say yes" behavior).However, the cultural models framework helped to dive deep into these aspects and identify its integral constituents and elaborate on those constituents' interactions. For instance, with respect to the Trust cultural

model, its instituted cultural element in the screenshots exposed the cultural values of these artifacts and meaning that they embodied. Furthermore, the ethnographic analysis facilitated me to expose how the cultural aspects were ignored (i.e., screenshot capturing steps missed) and what were the consequences of such ignorance (i.e., loss of trust and rise in clients' expectations).

5. The adoption of ethnographic-research approach has facilitated the exposure and explanation of some cultural models internalized by the vendor teams. The approach's outcome, in the form of the ethnographic narrative describing the cultural models, has provided some contemporary cultural-studies researchers with rich real-world cultural interplay illustrations to design simulated cultural training scenarios. For instance, Manosar designed and evaluated his training framework using GSE training scenarios that were designed based on the "thick description" resulting from my ethnographic investigation ([98]).

8.2.3 Feedback from Participant Organizations

I had the opportunity to share the findings and the preliminary analysis with the teams that I studied. The goal of share the findings and analysis were two fold: I was interested in (1) confirming whether my interpretations were correct and the participants were able to relate to the findings and (2) creating awareness about the subtle but significant role that the socio-cultural aspects were playing in the GSE (more specifically GST) practice. To this end, the findings were well-received and appreciated by my participant organization members, which is evident in the feedback that I received from some key members from the organizations.

Study S1. The delivery-management level participant from Vendor V1 emailed me mentioning that:

> (**S1.DM1**)
> Your observations on behavior[al] aspects on the QA [Quality Assurance or

> Testing] team was quite enlightening, and will help us in improving/streamlining the process.

Study S2. Again, the response to the studying findings from the head of the GST-equivalent division at Vendor V2 organization was:

> (**S2.DM1**)
> I do think this will be insightful and useful not only to those who are new to [GST], but even to 'matured' [GST] users. This would also be useful for vendors providing these services to get a different view of things [and so] I want to share this with the Test Leadership at Vendor V2.

Test Leadership was the group at Vendor V2 that focused on designing new solutions to improve the testing services they provided to their clients.

Study S3. Finally, at the participant organization Vendor V3, the findings were appreciated by three separate departments—the technical department that provides the GST services to clients, the research department that performs software-engineering research to improve the practice, and the training department that conducts training for socio-cultural awareness. Additionally, they expressed the desire to collaborate to explore the work in this new directions that emerged from the findings. In an email response received from a Delivery Management Level manager:

> (**S3.DM1**)
> [Hina's work was] well appreciated in [the vendor organization], in all [the] three groups, [Technical department], [Research department], and [Training department] have expressed interest [in pursuing future work in this direction].
>
> We will need to figure out aspects that make her work more than training, socio-cultural view to risk management, role of her work in large program management/relationship management.

8.2.4 Conclusion

As demonstrated by this research, the cultural models system has been a valuable tool towards exploring the culture's influence on the GSE practice. It has been instrumental in shaping a new discourse that internalizes culture beyond the static dimensional system.

My dissertation has illustrated how the cultural-models approach facilitated a deeper investigation of this issue of cultural influence within the GSE practice perspective. Moreover, the identification of the embedded cultural models, illustrations of their interplays, and discovery of newer interpretations of the commonly reported behaviors in the GSE practice, which was doable because of the cultural models approach, has open new horizons for additional systematic investigations in this direction.

These contributions can benefit both the researchers and the practitioners. For instance, the framework can facilitate other researchers to design additional studies and surveys to understand other aspects related to culture and global software engineering. Also, such cultural models can raise awareness among practitioners about culture's role in practice and serve as input for designing solutions such as cross-cultural training programs at the organizations. Such awareness and programs can greatly contribute towards an improved relationships between the clients and vendors leading to more profitable and successful partnerships.

8.3 Implications

This piece of research has spawned new directions for investigation. For instance, follow-up client-side research, similar to the ones I conducted at vendor-side, can significantly benefit to identify the embedded cultural models in the client side practice. Such an understanding can facilitate comparison of the models on both the sides, rendering new solutions to the cultural problems of the practice. For another instance, additional studies are required to be designed and conducted in a similar fashion to examine whether my findings are transferable to other similar settings. For yet another instance, the corpus amount of data collected during the study can be a rich resource for further analysis of data for gaining understanding of the GST practice from different perspectives. I plan to continue the data-collection (if and when permitted) and analysis to uncover other interesting cultural dynamics and hidden aspects of the GST practice.

In addition to the general areas for future work, based on my analysis and findings, I have identified some specific areas for future work for both researchers and practitioners, which are presented in this section in the form of implications for research and practice.

8.3.1 Implications for Research

The primary goal of software-engineering research is to enhance the software-development process and the resulting software products. The research concentrates on improving aspects such as the quality, efficiency, and effectiveness by developing new techniques and designing novel methods and principles. More recently, the research has directed its attention to investigating non-technical aspects of the existing practice to identify improvement areas. My research adds to this genre of research, where I investigated the cultural practice of global software engineering to identify future areas of improvement that the research can address. In this section, I discuss these future directions for research.

8.3.1.1 Developing Culture-Focused Testing Methods

Research has designed methods and strategies, such as user-centered design and usability testing, to incorporate culture's influence on the design and usability testing aspects of software development. However, similar research is lacking for the broader software-testing domain. Based on my research, I speculate that there is a need for investigating the subjective and culturally-influenced nature of software-testing. I reflect on the need to investigate concepts, such as culture-focused testing methods. I am not the first researcher to propose such an idea. Sommerville and Rouncefield proposed a similar concept—customer-focused testing [128]. They argue that the testing resources are limited and the software specifications do not precisely represent scenarios that are important from customers' perspectives. Thus, there is a need to investigate the use of ethnographic analysis to derive customer-focused test scenarios [128].

Whereas customers form one important parameter for deciding testing scope and coverage, there are other parameters that contribute to the decision-making process, such as the

environment, the software developer's culture, and client's/user's national culture [17, 120]. Hence, there is a need for extending the customer-focused concept to a culture-focused concept that includes these parameters. It is crucial to explore this concept because it may be surprising to find the profound influence of culture in the (global) software-engineering context. For instance, I observed that a culturally-different perspective taught the Japanese Vendor (S3T3) Team to become better at testing:

(S3T3.TM1)
Now, I'm not in a position to say that they have an advanced set of test cases because having gone through that [experience of understanding how Japanese client's testing team works] myself I feel that this is what the product should also be tested for. But at that point of time I definitely did not test each of these scenarios. And I still thought that I had sufficiently tested it [the software] and I did ship it to Kensa [Japanese client's testing team]. But then some of these things [tests] did not pass. Since looking at the requirements document I did not anticipate that they will test the code in such a situation also, I didn't test for it . . . our assumptions were very different from the requirements. (Quote 38)

The statement "since looking at the requirements document I did not anticipate that they will test the code in such a situation" indicates that the current method of testing based on requirements documentation has some drawbacks [128]. Cultural knowledge results in a set of assumptions and, as noted above, the Japanese Vendor (S3T3) Team's assumptions differed from the Japanese Client (S3) Team's expectations. This difference in assumptions resulted in failure to include the relevant user scenarios in their testing. Thus, failure to acquire and adapt to the client's/user's cultural knowledge can result in a failed effort in improving the quality of the software. This example reinforces my previously stated idea of investigating the concept of building culture-focused testing methods and techniques. Definitely, this idea needs more exploration and investigation.

8.3.1.2 Discovering Appropriate Team Structures

The study findings provide evidence that the team structures significantly influence the testing activities at offshore locations. Thus, research should concentrate on conducting studies

that investigate inter-organizational team structures. Some areas of research include investigating (1) whether my findings can be generalized to other teams, (2) whether importance is given to team structure before setting up GST teams in industry, (3) whether having vendor-side onshore teams that work closely with clients support or hinder the quality of testing activities, and (4) which type of team structure would be the best for reducing the pressure and, thereby, facilitating a practice that produces high-quality work and product.

8.3.1.3 Designing Cultural-Training Frameworks

I observed that there is lack of research in the area of designing culture-sensitivity training programs in global software-development settings. I speculate that the primary reason is the research community's unawareness of how subtle cultural factors are having a surprisingly significant impact on the practice. Although, large organizations have some culture training programs that are informative, they emphasize only the visible aspects of culture (e.g., food, handshake, and etiquette) as discussed in Section 8.1.2. What is required is to revise these training sessions to incorporate cultural sensitivity regarding the invisible aspects of culture (e.g., thought process, problem-solving strategies, and values and perceptions) that affect the software-development practice. Thus, the research community can facilitate the *design of cultural-training frameworks* for (global) software-development settings that can be used by the practitioners to create cultural-training programs.

8.3.1.4 Translating Cultural Impact to Business Value

"If business people want to gain understanding of and allegiance to their corporate goals, policies, products, or services, wherever they are doing business, they must understand what those and other aspects of management mean in different cultures" [140]. As research suggests, cultural values and knowledge intertwine with the business goals and missions, resulting into practices that are inherently culturally influenced.

Although it is widely known that culture influences business [68, 140], to the best of my knowledge, research lacks at defining agencies that can translate culture's impact into

business value. The lack of such agencies might be the reason why cultural issues are not gaining the importance that they deserve. As I demonstrate in this research, the various cultural models and their interplays resulted in a narrow escape from a project failure (Mexico project). This project was $1.5 million worth, with the potential of bringing in more revenue. However, the cultural troubles were not identified, perceived, acknowledged, and gauged in terms of business value and loss. Because the project succeeded eventually, thanks to yet another cultural "survival" model—flexibility—that came to the rescue," the experienced troubles were disregarded. This demonstrates the lack of acknowledgement to the role of cultural system in creating as well as resolving business challenges.

Thus, such exemplars support the need for designing agencies to measure business impact. These agencies may be as simple as a study design: for all the failed or almost-failed projects, investigate the reasons for failures and sum the project costs for those that failed because of cultural issues such as the one described in this research. It might be surprising to see that many projects undergo similar troublesome experiences. The speculation of such an outcome is based on the information shared by my participants and members of audience (who attended my talk when I shared the findings with the organization): they all agreed that the cultural interplay pattern presented in this research was indeed a common pattern experienced by them on a frequent basis. If such troublesome experiences are occurring on frequent basis and they are still going unacknowledged and unnoticed, it implies that the system is broken somewhere. Translating the experiences in the form of business value and loss might be a hopeful way to seek the attention that cultural systems deserve.

8.3.1.5 Investigating Other Cultural Models

The ethnographic analysis I did using the cultural models framework helped uncovered unique cultural models such as Flexibility Model that are embedded in the vendor-side of global software practice. As illustrated in Section 6.3 such models play a significant

role in the success of global software practice and business. On the contrary, using dimensional approaches to investigate cultural impact have failed to uncover such specific cultural knowledge systems because of the limitations of such approaches (discussed in Section 2.3). Hence, there is a need to adopt more dynamic approaches such as the one demonstrated in this research to investigate other aspects of the global software engineering practice. Moreover, it will be interesting to explore whether cultural models such as flexibility coult be the next set of dimensions that can be augmented to the existing list of Hofstede's dimensions.

8.3.2 Implications for Practice

Based on the research findings, I provide the following recommendations for industry to consider and explore further.

8.3.2.1 Appreciating Test Engineers' Efforts

As described in Section 4.1.7, test engineers are in a unique situation in the organization because (1) testing jobs have the property of generating more work on doing work, (2) exhaustive testing is not possible, and (3) test engineers are considered as second-class citizens. These factors seem to create reduced motivation and enthusiasm levels among the testing teams. Because the test engineers are the ones responsible for assessing and ensuring good quality of the product, it is important that the quality of their work is noticed and appreciated not only by the client organization but also by the vendor organization.

Based on my findings and analysis, I believe that high motivation and, consequently, high quality can be achieved in several ways.

First, I found that the participants perceived their testing profiles as important, with the responsibility of delivering high-quality products. This perception, which indirectly also represents the perception of the organization towards the test engineers, contributes significantly towards higher motivation levels among test engineers. This organizational practice should be recognized and promoted to ensure continued benefits. Second, situations where

test engineers experience the dilemma of meeting deadlines by compromising work quality should be recognized, investigated, and appropriate actions must be taken to avoid such situations. Third, practices may be implemented to encourage the test engineers (and in general, other employees in the organization) to be innovative in their work (e.g., allowing employees to spend 10% of the work time per week for implementing innovative activities and rewarding the best innovative ideas). Finally, organizations can start recognizing and rewarding the other roles, such as information providers (see Section 4.3.2), that test engineers seem to play in the organization. Because test engineers have relatively more time to understand the requirements than the developers (during the development phase and while developing the test cases), often they have a better understanding of the requirements than the developers. Hence, under crucial time pressures, the developers sometimes consult the test engineers for clarifying their understanding. Recognizing, promoting, and rewarding such roles can help motivate test engineers to enhance the quality of testing and service that they provide, and increase the efficiency of their testing tasks.

8.3.2.2 Acknowledging Quality Dilemma Situations

Ample evidence that emerged from the study indicated that the participants strove to maintain their quality of work and, consequently, the quality of the product and the service they delivered. However, they were faced with challenges while attempting to achieve and maintain high-quality work. Overemphasis seemed to be given to numbers and productivity seemed to be measured mainly in terms of numbers. Other factors such as complexity of tasks were being ignored. Consequently, under high time pressure (leading to high work pressure), many factors led to compromise of the quality of the testing.

Therefore, to ensure high-quality testing, it is important to appreciate and acknowledge the situations under which testing is being conducted. Also, it is crucial to identify factors that result in quality dilemmas so that appropriate measures can be taken to avoid them. In addition, methods and metrics should be investigated and implemented that emphasize

task complexity. Additionally, the complexity-related information should be accordingly conveyed to the clients so that decisions about productivity measures can be made by considering the complexity information too.

8.3.2.3 Understanding Client and Vendor Cultural Models

Investing effort early (e.g., when service-level agreements are finalized) to understand differences in embodied meanings of various work-related concepts can significantly benefit both the organizations. First, such an understanding will provide awareness among the two teams about the cultural differences in their work styles so that effort can be directed accordingly to enhance work-related compatibilities. Second, this understanding can facilitate the implementation of strategies for avoiding the unproductive-productivity experiences. I speculate that avoiding such experiences can lead to more productivity and less boredom for the vendor teams and cost savings for the client teams by avoiding the execution of unproductive tasks. Finally, understanding such differences (which result because of cultural differences) can help the organizations attract, recruit, and, more importantly, retain highly-skilled employees. For instance, cultural differences between clients and vendors sometimes result in unpleasant circumstances, which makes it difficult for the members of the two teams to work amicably. This appears to result in high attrition [31]. Investigating correlations between (1) the the client's and vendor's cultural differences and (2) and the attrition rates at the two organizations can help in formulating strategies for attracting, recruiting, and retaining skilled employees.

8.3.2.4 Improving Cultural-Training Programs

There are several implications for the organizations to improve their practice by better handling the cross-cultural dynamics at the organizations that have emerged as a result of my study.

Creating cross-cultural awareness. First, there is a need to invest effort to *create cross-cultural awareness* among practitioners of both teams that are involved in global settings (i.e., the vendor as well as the client team) because it can be beneficial in several ways: (1) reduced confusion/delays among delivery teams due to mis-communication; (2) better engagement with the global-team members; (3) enhanced understanding of expectations across teams; (4) reduced costs by achieving project objectives, time lines, and budget targets; and (5) enhanced experiences resulting in better productivity and quality. Awareness can also be used to predict potential clients' behavior (e.g., uncertainty avoidance factor) and expectations, which can in turn help make better effort estimations.

Evaluating cross-cultural initiatives. Second, although organizations conduct training programs to bring cultural sensitivity to their employees, there is a need to *evaluate cross-cultural initiatives*. Based on my findings and analysis, such programs touch on the high-level concepts of culture (e.g., handshaking style and business etiquette). Hence, the practitioners appear not to view such programs as adding any value to their work because these concepts do not directly impact their daily work routines. The industry should make use of research findings to evaluate and, if required, change their programs to demonstrate the deeper impact that culture has on their way of executing their practice (e.g., mismatched perceptions of quality). Such an effort can enhance the business value of testing because the practitioners' can better connect to the subtle cultural difference with their clients, and direct effort towards working on aspects that are of most value to their clients and their business.

Managing cultural knowledge. Third, there is a need to *manage cultural knowledge* produced through client-vendor interactions. Large vendor organizations, like the one I studied, have the opportunity to work with many clients from different cultures. Such organizations acknowledge and support customized ways of working based on clients' preferences. However, there is a need to design strategies for understanding and synthesizing

the cultural learning and knowledge gained from working with various clients. Not only will such an effort benefit organizations by helping them to better cater to their client's requests, but it will also facilitate their providing better service to the other clients. Leveraging the diversified cultural experiences gained through working with different clients will significantly benefit the organizations for strategizing their business capabilities.

APPENDIX A

STUDY I GUIDE

1. How many years of experience do you have in total?
2. Since when have you been in testing?
3. How do you perceive testing? How do you view your role as a test engineer? For instance if you were a tester, how would you consider your role to be?
4. What motivates you to work on testing project? What de-motivates you?
5. What do you think are the advantages of being in a testing team? What are the disadvantages of being in the team?
6. Can you narrate an incidence when you found testing very interesting? What about very disinteresting?
7. What aspects of testing do you like the most? What aspect you don't like at all?
8. How does the organization perceive testing? How are test engineers considered at the organization?
9. It is quite known that testing is considered to be a second class citizen job? What do you think about this? Why do you think people think like that?
10. When testing application a number of times what strategy you use to decide what to test and what not to test?
11. Do you frequently interact with the development team? If yes, can you tell me more about how the interaction goes? How do you communicate with the development team? Can you narrate specific incidence when you were communicating with the developers regarding testing a bug?
12. Any instance when you were not able to produce good productivity while testing? What do you think helps you to give a better productivity in testing? What affects your productivity in testing the most?
13. How is the work environment here? Can you narrate a specific instance when the environment

affected your productivity in testing?

14. Who is responsible for certifying the application once it is tested? Is it a very responsibility oriented task? If so, what kind of responsibility is involved?
15. Communication Related
    (a) Whom do you communicate with in case of doubts while testing?
    (b) Why do you first verify the bug with the testing team and not log it directly? (Question designed based on what I observed)
    (c) How is the communication with the teams?
    (d) What kind of rapport do you have with the teams? (Get more information about their rapport with the development teams)
    (e) What documents are shared by the other teams with you? How?
    (f) Can you narrate a specific instance of communication, when communicated with the development team?
    (g) How do you clarify whether something is a defect or functionality?
    (h) What issues do you frequently discuss with the development team and the functional teams?
16. In your entire experience, is there anything specific that you found very challenging regarding your job?

APPENDIX B

STUDY II GUIDE

Opening Script

- Introduce the researcher and give a brief overview of the research project.
- Discuss the way the interview will be conducted and the notes will be taken.
- Go over the confidentiality and privacy concerns. Explain how we will anonymize the data.
- Ask partcipant's permission to record the session.

(Start recording now. Depending on how the conversation proceeds, the questions in various categories listed below can be used as prompts.)

Grand Tour Question

- I have known that testing is a very interesting activity but have never experienced the role of a tester. Can you tell me something about your experience as a software test engineer?
- Can you tell me something about your role in this project?

Perception

- What do you think about testing as a career?

Deadline Pressure

- As I have never experienced a deadline pressure situation for testing, can you describe what a deadline situation is like? What situation would you consider as a deadline pressure situation?
- Which situations typically creates immense deadline pressures? What factors, do you think, result in such deadline pressures?
- What was the last time you experienced such a pressure situation in testing? What created the situation?
- Can you tell me about a situation when you were put in a position where it was impossible for you to finish the assigned tasks in the given deadline? How did you manage this situation? How do you manage your overload tasks under deadlines?

Coping Strategies

- Given such deadline pressures, how do you go about accomplishing these tasks? What strategies do you use to manage your testing tasks, under deadline pressures?
- How do you go about deciding the priorities of work under deadlines? What factors determine which strategy to use for priorities? How do you go about selecting the tasks?
- What factors do you think were very beneficial for surviving through the deadline situations?
- Are you satisfied with the current strategies you adopt to handle testing under pressure situations? If not, what do you think should/can be changed/improved? (follow-up question)

Manager's Response and Support

- Can you narrate an incident when you were content/satisfied with the manner in which your seniors handled the deadline pressure? What factors contributed to the satisfaction?
- Can you narrate another incidence when you were dissatisfied with the manner in which your seniors handled the pressure? What do you think was missing or wrong?

Quality (If they talk about quality)

- How do you think these deadline pressures affect the quality of testing?

Communication and Information

- Do you directly communicate with the client team? What kind of interactions do you have with the client team? Once you know about the deadline, what preparation do you do for the upcoming deadlines?
- Could you tell me a time when you had trouble finding the information you needed to meet the deadline and it was not available to you? What information was it? How did you manage to get that information? What did you do when the information wasn't available?
- How much visibility do you think you have about upcoming deadlines? (*For managers:* How do you share this information with your team members?)

Team Dynamics (If they talk about team dynamics)

- Have you observed any specific team dynamics changing under deadline pressure?

Conclusion

- If you were to list a set of things that you learned through these experiences while handling testing activities during these immense deadline pressures, what would those be?
- Is there anything related to testing in this interview that you would like to say that we haven't covered yet? Are there any questions that I didn't ask that you think I should have asked?
- Do you have any questions for me?

APPENDIX C

INTER-RATER RELIABILITY REVIEWER'S GUIDE

Introduction

This study is part of an ethnographic research that aims at understanding the how culture plays a role in the global software-testing practice. The qualitative data set to be analyzed in this research includes a set of transcriptions of interviews, observation field notes, informal discussions, and artifacts and other documents. The research uses thematic analysis approach with grounded theory coding as a basis for performing inductive analysis. To validate the coding and analysis processes, you're being asked to participate in a series of sessions involving coding (individual), discussion/negotiation (collaborative), and evaluation (collaborative).

Procedure

The attachments will contain a transcript and my coding system. For validating my coding system, I request you to do the following

1. Go through my codes and understand their meaning using the definition and the examples. This will also help you better understand the focus of my research
2. Next, go through the transcript and code the data independently. You may refer to my coding system to use codes that, you think, aptly describe the context of the conversation or you may create your own codes.
3. We will set up a date and meet and discuss our set of codes. We will negotiate over the coding to understand the differences, and note the level of agreement/disagreement.
4. If there is more than 70% disagreement, then I will recode my data based on the discussion and feedback received from you and we will repeat step 3 and 4 until we reach70% agreement.

Research Study and Participant Details

The participant group can be divided into the following group

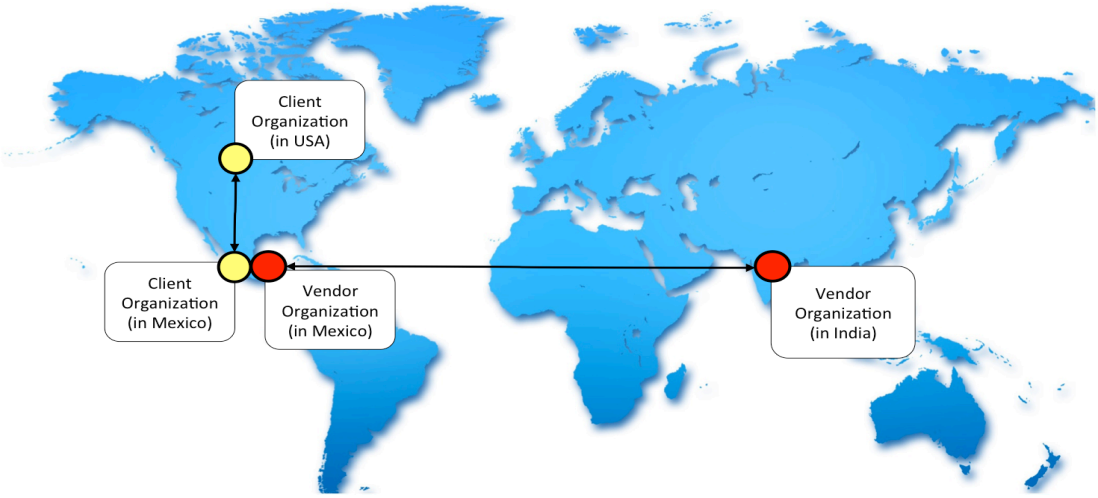
1) **Indian** participants working at **offshore vendor organization** (INFY) in India
    a. for **U.S.A. clients** (KR)
    b. for **Mexican clients** (KR)
    c. as market researchers
    d. as a part of the cultural training group at the organization (who is responsible for conducting cultural-awareness training)
    e. board-level management members of the organization
2) **American** client team members from **client organization in U.S.A.** (KR)
3) **Mexican** participants from vendor organization (INFY) working as **onsite coordinators** in Mexico

<u>Team Structure</u>

The team structure is as described in the diagram. The team studied consists of the offshore vendor organization, the onsite vendor organization in Mexico, the client organization in Mexico,

and the client organization in USA. Below, I explain some common terminologies used in Global Software-Engineering setting:

1. **Offshore Vendor team –** the team that is located remotely as compared to the client team and provides the software-development related service to the client organization.
2. **Vendor and Client team –** the client team is the one who is seeks the service that is provided by the vendor team (i.e., the service-provider team).
3. **Onsite coordinator –** Vendor team members who works onsite (at client geographical location) and are responsible for the collaboration and communication between the client team and the offshore team



Research Question

The research focuses on the following research question

> *What and how do cultural factors influence the way global (offshored, out- sourced) software testing is practiced and what cultural models are embodied in this practice?*

Some sub-research questions that I attempted to explore are

1. What cultural meanings are the vendor software-test engineers using to organize their behavior and interpret their experience with the client team members?

2. What tacit assumptions do software-test engineers make culturally when performing their daily activities?

3. What is the culturally embodied meaning of software testing for the vendor team members (i.e., general notion/thoughts they carry about testing)

4. How do the test engineers acquire cultural learning of their clients at the vendor organization?

5. How are t he global software-testing activities perceived by the vendor teams (i.e., the way the participants practice testing activities versus what seems to be expected)?

6. What interpretation/understanding do the vendor teams have of the client team's expectations? How are these expectations (the understanding of the testing tasks) conveyed, communicated, and negotiated between the client team and the vendor team?

7. What is the intended software-testing practice agreed upon between the client team and the vendor team? What is the current practice observed at the vendor end? Is the current practice different from the intended practice? If so, how?

APPENDIX D

SAMPLE TRANSCRIPT

1	03 Apr - Followup with SBen
2	**Infy: Vendor (offshore) organization in India**
3	**SBen: Vendor-side project manager in India**
4	**KR: Client organization (it represents the Mexico (MX) and the USA organization)**
5	**Process team: Client's business team, which represents the client's team**
6	**LE (LE-client): Client side manager from USA, who is responsible to oversee the Mexico KR team's business as well**
7	**Csr: Vendor manager in Mexico**
8	**HUL: KR MX client manager**
9	**VJ: Vendor onsite manager (Onsite managers are responsible to interact between client and the offshore vendor teams)**
10	**Context: his interview was conducted right after the weekly status meeting meeting, in which the vendor team (Infy) updates the USA client manager (LE) about the testing status of the Mexican client project (MX-KR project)**
11	**Purpose of this interview: In the weekly-status meeting, I observed that the MX client team wanted the testing activities to be planned in a way that they would perform much of the testing in MX. This means that the vendor team at offshore location would not be performing much of the testing. This had surprised LE because they have the outsourced, offshoring setting to avail the testing service from the Infy team. In this case, the MX-client team was not utilizing this service.**
12	
13	Friday, July 19, 2013
14	2:57 PM
15	[HS]: can you explain me what was going on (in the meeting) in the sense of the numbers and what LE was saying?
16	[Sben]: see basically the way it works is that there is a KR process team… Those are people from KR or [another Vendor organization] and all. They are the process team people and the other people from the other side (client side). So they are also doing the testing and we are also doing the testing [00:25]. So the understanding was that any new functionality, they will tested first. Or they will have to test… That means it's a mandate for the process team that if any new functionality is coming in any cycle they have to test it. And we will test any existing functionality which is not getting changed. And there are two folders or kind of environments (in which software testing will be performed). One is the official folder and the other is the training folder. So what we as Infy people are supposed to do is that in each cycle we discuss with the process team people, because finally they are kind of in charge of this whole testing. So we work with them and we decide the scope for official folder. Official folder is the folder from which all the status reporting to the senior management etc. happens. So this is the scope for the entire cycle three okay. [Starts explaining the table as shown at the end of the file.]
17	
18	In addition to the status report sent by Csr, please see below scope for Cycle 3. This may change slightly, since for some of the areas discussions are going on between our Mexico team (i.e., vendor team in Mexico) and process team. nfyteam would also re-execute the test cases under process team scope in training folder.
19	

20	So now this is the scope of the entire cycle three. So total 1431 test cases. This is each track..OTC, MTI, etc… In OTC in official folder this time process team will be executing 47 test cases and we will be executing 33 test cases and so on. And generally what we do is we start with our scope, Infy side. And once we finish our scope, whatever process team is executing in the official folder we reexecute that in training folder. So it just becomes kind of revalidation; in case processed team has missed any defect or something then we can catch it. So far, two cycles have already been completed. And now we will start the cycle three. Now why LE was raising concerns is that she was thinking that in cycle 1 some new functionality came, so process team tested that and we tested the existing functionality. Again in cycle two some additional functionality or some additional interfaces came, so process team tested that and we tested only the remaining scope in a special folder. Now she is thinking that in cycle three there should not be much new functionality. So ideally only we should be testing everything, process team should not be testing anything in the official folder [02:36]. But still if you see the process team is saying that they will do these testing…so she is asking why then Infy is not doing the entire testing, why process team is doing half the testing we are doing half the testing. So what process team is saying is, although some of the scenarios are there in the cycle two, they still want to reexecute in cycle 3 with some different data.
21	
22	On that VJ said that over all HUL (he is a KR MX client manager) is in charge of KR Mexico. So, VJ was telling LE that someone like HUL should agree that Infy will do entire testing, process team need not do any testing in this cycle. But that HUL has to take the stance we cannot take the stance [03:23] because we cannot tell the process team that you do not do any testing, these entire 80 test cases we will only execute. Because currently the process team drives this (planning activity). They are given the authority to agree on this scope as to what they will test and what we will test. And that's why she was telling why so many numbers are still under process team. Ideally, she was expecting that the majority number should be on our (vendor) team.
23	
24	[HS]: did you ever propose to the process team that we can do all the testing?
25	
26	[Sben]: Yes, right from the beginning we have been saying but see finally what happens is that the process team is majority KR people. KR high-level business kind of feels more confident if their team does the testing. It is just the kind of comfort factor because Infy is finally any third party vendor for them. So they have more obviously…[He uses the word "obviously" reluctantly] in fact so far… In fact… In fact this is the first time in Mexico that Infy is doing the testing. So far entire testing used to be done only by the process team, in Mexico. So this is the first time they said that okay Infy can also do the testing but… That's why they did not want to give the entire team to us because they did not feel that much confident. So they said that in any testing cycle any new functionality that will come… Because generally new functionality is more critical. That (new functionality) may generally have more defects. Anything which is an existing functionality, which is just going on a rolling out will be more stable. So because they wanted to have some kind of more confidence, they said that in any cycle any new functionality we will test. You only test the existing functionality. So maybe slowly they want to transition. So that's why in cycle one, they tested every new functionality and we tested existing functionality. And whenever we got the bandwidth (they use the word bandwidth in the sense of "approval to perform a task" or "time") we tested the new functionality. So now LE was expecting cycle three, all the new functionality should have

	already been tested by us (vendor team), and so in cycle three we will do all the testing. But this process team is saying no. They are saying that these (new) test cases we will execute, remaining you execute. So that's where the conflict is, where VJ said that this is something…Csr…he is our Infy person at on-site (in Mexico). As VJ was telling Csr he can again talk to HUL, was overall in charge KR MX and check with them if he is okay and that we do most of the testing in official folder and process team should not do much work [05:14]
27	
28	[HS]: so in cycle three, is there a new functionality that has been added?
29	[Sben]: some new functionalities also there although it's not a new functionality. What had happened was some of the interfaces… Means this application interfaces with other applications… So some of these interfaces were not there in cycle two. These have come for the first time, especially for OTC. [06:10] because of which may be process team might be be wanting to test on their own {HSAnalysis: he's uncertainty language indicates that he is making an assumption or guess here.}. Because this is also for them it is something new…
30	So I didn't speak at that time because I did not have the exact number…like out of these 47 how many are really such cases and how many other repeated test cases. So that is something that Csr can talk to the OTC and find it out. But otherwise in cycle three there is nothing new functionality as such coming up. That's why LE was saying that why is process team doing so much of testing…. [[06:14].
31	
32	[HS]: and especially STP has 692 test cases?
33	[Sben]: but here the problem is… That's why VJ was saying that there is something known as readsoft. Means there are many test cases for which there is a step known as readsoft. It is a kind of device with which you use and scan the invoices. So it is a kind of of physical activity… And that machine we don't have here. So that only they have. And so that is something that anyway they will have to do… because that machine we don't have here. And so because of that that count is looking high, we haven't been given access to that machine. Any option we don't have here and at on-site also they are saying that because they have the machine only they will do it. So because of that the count is higher for them. Means still we do some of the steps in that test case, but readsoft related test cases are marked against them. Because unless readsoft is done that test case does not get completed.
34	
35	[HS]: but then we have a big team of Infy in Mexico right?
36	[Sben]: not big, only six people. And a lot of their time goes into coordination. So we have almost 30 people here. So on daily basis we do the execution, we have a lot of queries, which KR may have to resolve. We send our queries to resolve the issues to our on-site coordinators; they then talk to the various KR people and resolve it for us. But they don't get any real execution done there because they don't have the bandwidth (i.e. the time). But still we had shown the willingness that give us access to the readsoft they will do something. But still they did not. Maybe logistics wise it must not have been possible. [08:20] so I don't know physically where that machine is and where our people are. So all those issues might also be there.
37	
38	[HS]: so and one more thing is that you know you remember you had last and explain me the cycle 1, Cycle 2, cycle 3 concept. [08:28] so basically what you told me is that in cycle one most of the functionality is tested, that the bug fixes happen and then the cycle 2….but over here now it looks like there is a different scenario. they are adding new functionality in cycle three...

39	[Sben]: such things happen in any testing project. So the idea is that we do 3 cycles. They try to make maximum functionality available in the first cycle. But rarely it happens in any project that 100% functionality will be available in the first cycle. Although that is the target, many a times some development gets slipped and all. So, by the time cycle one starts only 80% of the functionalities available. So we do that – 80% testing. Now the remaining 20% should be available in cycle two but sometimes it may happen that in cycle two also the development team may not have finished the remaining 20%. Maybe they might have finished only 15%. So, so the bottom line is that in cycle two we have 95% functionality available. Again we do the entire cycle two testing, so again we will test everything plus from cycle one we would have found certain modules in which more defects are there, we will retest those in cycle two… So that was having several days now. So similarly in cycle three then the entire functionality is available we again retest and we redo the critical modules. Now here in cycle three again testing everything. But in many projects, what happens is that cycle three is a catch-up cycle. Means that if all the functionality becomes available in cycle one then, then give a test once in cycle one, the testing cycle 2, and cycle three is many a times optional in many projects. Only if they find many defects in cycle two, they will do cycle three otherwise they won't.
40	
41	[HS]: and you mentioned that they're going to do everything (i.e., all the testing) in cycle 3. You mentioned that you want to do half part of it right.
42	
43	[Sben]: we mean between process demand as we are going to do everything (??)… So in each cycle we are executing everything. The only thing that is happening is,in cycle one some of the functionality was not ready. So that got available in cycle two. And in cycle two most of the functionality was there but couple of interface-related applications were not ready. So in cycle 3 those interfaces related applications are the first priority. So in cycle two what we did was whenever those interfaces were there…. So because the interface application was not actually there…. We kind off simulated… Like a dummy kind of a value we put in just like a for a workaround… And we completed the test cases. Now in cycle 3, they will be testing with the actual interfaces.
44	
45	[HS]: will these parts again have the cycles?
46	[Sben]: no no…
47	[HS]: so one module will be tested only once?
48	[Sben]: ideally this should not have been the case. We have highlighted these risks to the clients… That when we realize this in cycle two that those interfaces will not be available in cycle two, we highlighted this risk that if those would be available in cycle three then those are going to get tested only once. So businesses is aware of that… About such risks. [11:36]

OTC		**MTI**		**RTR**		**STP**		**Total**
Process Team	Infy Team	Process Team	Infy Team	Process Team	Infy Team	Process Team	Infy Team	
47	33	108	160	0	270	692	121	1431

Team structure diagram

APPENDIX E

CODING GUIDE

NOTES:	[HS] refers to the researcher Hina Shah, who conducted the data collection [KR] refers to the client organization [Vendor V3] refers to the vendor organization

Code System	Code Description	Data Collection Procedure	Sample Quote
Approach towards a Problem and approach towards solving it	This category captures the approaches used by different teams towards problems they encountered and solutions they came up with to resolve the problems	followup interview after observation	[S3T2.PM2] (Vendor Team): Ok, for (test) planning...we always plan10-15% in the first week, 20% in the second week, slowly gradually increase, so by that time the environment stabilize the data issue is resolved, so we get the speed. It should be a curve; it cannot be a flat projection. We tried to do that curve projection but the customer pushed for this flat projection. They said "no we need everything" and we are not able to convince them. We were not able to pushback on them saying that this is not the right way of doing it and that is where the problem was. There with my onsite team because of the different culture because they were new, they were all Mexican folks, they were issues in terms of onsite-offsite communication. Lead were not able to convince the customer that this was not the right way of doing, so that was one of the reason that we over committed in the first two weeks and couldn't deliver, definitely it's not right. That was only definitely the big issue.
MX: Provide Available Information Immediately	MX team's approach towards the problem was to provide the information immediately when it becomes available.	followup interview after observation	[S3T2.TM1] (Vendor Member): So this time, in Cycle 2 itself it started. The major discussion what we were having was that, that phase should not come. [HS] (myself): Again in the Cycle 2. [S3T2.TM1][23.07]: Again in the Cycle 2, so that's why we were trying and this. [HS]: So what exactly did you do? [S3T2.TM1]: Haa, so now we are having a solution. [HS]: What's the solution arrived? [S3T2.TM1]: The solution arrived was like in first Cycle 1 – what the Process Team has done, they had released the "Product Costing" for all the Plans on the same day, for all the Plans. Now what they did is they also Planned for the "Product Costing". And they are going to release the plans whatever plans get completed first. I mean they have given that there was one specific plan they will release by today and we can start up from tomorrow.
Ven: Relying on Flexibility	Demonstrating the flexibility offered by the vendor teams in terms of planning the testing schedule (moving around test cases in weeks), working over the weekends to finish the targetted tasks.	followup interview after observation	[The problem here was that in cycle 1 of testing, many things went wrong including testing planning and execution, which resulted in slippage in test schedule] [S3T2.TM1] (Vendor Member): So that is one thing from Process Team perspective. And I also, I mean I asked ER-Client to interact with the (client) team, the test cases which we can work out from Week 2 to Week 1. So now we have got that, which all things we can start up from tomorrow. So this will definitely reduce our work pressure in Week 2. Simply it's like my Team is sitting ideal for 1-2 days. And simply they are coming on and staying back a long and working over the weekends for the next week. So that is not feasible, that's what we are trying to tell them that if you can help us, we can utilize the time, which is there.
Approach Towards Testing	This category descrbies the attitude, perception, and inclinations of the team members (vendor and client) towards testing as a practice		

Code System	Code Description	Data Collection Procedure	Sample Quote
Japanese Organization's Approach Towards Testing			
Focus Additionally on Environment	Describes the thought process of the Japanese to not only focus on main functionality but also on the environment and assumptions made around the environment.	Unstructured (ethnographic style) Interview with participant	[S3T3.TM1](Vendor Member): but for Japanese, they will test whether the application is sending, but they will also test what will happen if the mobile is switched off, what happens if the server which is suppose to send the alerts itself is down, will it send the alerts when it comes up? Or what happens if the amount is $1000.01 yen or dollars whatever it is.
Process Focused	Orientation of the Japanese teams is more towards focusing on processes and extent of details required by the Japanese customers	Focus group discussion with the Market Research Team member and a vendor manager to understand testing service from the market research point of view	[VSh] (Market Researcher):my insights about Japanese customers is that they are highly process oriented. Whatever interactions I have had in IT, earlier on I was in a different industry, but from IT dealing with Japanese clients, they are extremely focused on how you can take the processes from level I to level II. How can you essentially help them to improve on the processes. So, they are more focused in terms of addressing the… Rather than the end goal, the root to reach the end goal. The processes make a lot of difference.
High Sense of Sufficiency	Refers to the extent of testing expected to be performed on the system that will make the clients think that sufficient testing is done.	Followup interview with participant	[S3T3.TM1](Vendor Member): The concept of sense of sufficiency I believe is very different. I have not been is not interacted firsthand with them. Probably it's the way they are structured. It's quite related to the a lifestyle. They… Whatever could be an advanced scenario for some of the other geographies [38:07] it's a given for the Japanese. So their sense of sufficiency comes at a very high level, when very few things are breaking. And they have very very high expectations. It's easier to reach a sense of sufficiency milestones if you are working for an American customer
Don't Make Assumptions	Japanese team's approach is to clarify every possible aspect and avoid making assumptions and they expect the same from their vendor teams.	Unstructured (ethnographic style) Interview with participant	[AM] (Vendor Member): they [Japaense clients] don't assume anything, they don't take anything for granted. So that leads to lesser issues I believe. Typically, what I have seen in India is that we take a few things for granted and typically those things go wrong...That is one thing I found different there. That they don't take anything for granted.
Effort Estimation Approach	Describes Japanese team's approach not to worry about the 8 hours limit but to consider how many resources are available and by when the task have to be finished, even if that requires extra hours of work.	Unstructured (ethnographic style) Interview with participant	[S3T3.TM1](Vendor Member): I'm not going to deliver that is anything beyond what is written in the requirements documents. But from what I have seen still the reaction from the onsite (client) site is that this is something that you should have anticipated. Y

Code System		Code Description	Data Collection Procedure	Sample Quote
	Focus on Root Cause	Japanese team's approach to concentrate of the root cause of the problem by conducting deeper investigations such as why-why analysis	Unstructured (ethnographic style) Interview with participant	[SN] (Vendor Member):Five why (or why-why) technique example: so say that if we found a bug in the code, why did this bug arise? Because the functional requirements were not clear, why would the functional requirements not clear? Because the requirements were not captured properly, why were the requirements not captured properly? Because the communication had a problem, why was that there is a problem with the communication? Etc. so they basically want to understand the root cause of the problem and ensure that the root cause of the problem is fixed.
Mx Organization's Approach Towards Testing		Describes the approach that the MX clients held towards testing including review, number of test cases executed, bugs found, and others.	Interview with the Mexican Onsite Vendor Team Member	[S3.OC1] (Onsite Mexican Vendor Member): And both. I think in this project time is the most important part. [HS]: The timing is the most important part? [S3.OC1]: Yes for this project especially. I even think that it is more important than the quality. They (MX client team) prefer time more than quality. That's my point here, only point here.
	Adamant Ways of Working	Lacked of flexibilty towards performing planned testing activities	followup interview after observation	[S3T2.TM1] (Vendor Team): It (Product Costing Details) has been released also but client was not allowing us to take it up from Week 2.
	Testing for training	Testing is seen as a medium to train the users to the system.	Group interview post client-vendor meeting observation	[S3.CO1] (USA Client Manager): the biggest concern I hear from my clients is around the business as far as they go. Some of them use the testing to do training. So they're afraid if I use a testing team they are going to lose that opportunity.
USA Organization's Approach Towards Testing		Describes the approach that the USA client team was adopting with respect to testing activities.		
	Focus on the Major Functionalities (USA)	Describes the thought process of the USA, which mainly focuses on major functionality first (being realistic)	Unstructured (ethnographic style) Interview with participant	[HS]: "difference in the level of thought" [raising the voice to indicate it's a question] [S3T3.TM1] (Vendor Team Member): in the sense, lets say if there is a product to be designed and delivered, so Americans would focus on the major functionalities: this is what I want from the product. List there are some success cases, I want this banking application to send SMS alerts to the users as a debit of more than any amount have happens...So their focus will be more towards is it [the application] sending it whenever a debit happens?
	Focus on Understanding the Cause	Emphasize on understanding the immediate cause of the problem not necessarily the root cause and do what is the basic requirement to address the issue.	Followup interview with participant	[S3T3.TM1]: They will also reason the vendor why this particular problem occurred. But they will not insist on going to five levels of why telling them that we do understand that this is the reason. They would want to have that sense of confidence that the vendor understands the problem, and that they will be able to correct it.
	Intuition, Judgement and Past Experience Based	Evidence demonstrating that some testing decisions were made based on member's judgements, which were influenced by past experiences.	Online Meeting Observation	[Attendee Manager] (Client Manager in a Workshop meeting): We're expecting changes to the forms. Many a times based on our past experience what I have noticed is that this becomes a major issue because all your forms come from the external vendors and if they're not reallythoroughly tested then, we will end up with a lot of issues.

Code System		Code Description	Data Collection Procedure	Sample Quote
	Being Realistic in Commitment	Expressing the need to be realistic when committing to the clients about the tasks that can be done at offshore.	Artifacts, Documents, and Chats Information	[From the Annual Feedback Provided By Client To Vendor]: "Estimates should include contingency time? Don't be overly aggressive. Be realistic. This includes small deliverables not just major milestones. Credibility is extremely important."
	Making Assumptions	Refers to the judgements made for test cases' selection based on some assumptions that the US team makes	Online Meeting Observation	[S3.CO2] (Client Manager): I assume that we wont be able to test this in cycle one. But with the active directory changes we don't need to be able to at some point set that the keynotes are properly set to the appropriate e-mail addresses…or is that something that we will just have to assume that it is set out to be correctly, and the X team will handle it correctly if they get the new e-mail address?
Vendor Organization's Approach Towards Testing		Organizational attitudes, processes, and practices that describe the vendor organization's approach towards testing (or independent validation service).		
	Customer Focused Testing	Describes how the vendor organizations approach towards testing is to focus on what the customer is asking for or what the customer needs and work accordingly to provide those services.	Interview with Senior Manager	[S3.DM2](Senior Vendor Manager): So the realization happened, and moreover if you look in year 2000 when we were on the .com era, lot of applications were intended to be used by the end customer [3:29]. They were not meant for internal use of IS, or within the organization. They were actually systems,applications, products that were launched over the Internet. So now the risks of having defects was extremely high. And therefore this is the whole genesis of what we called independent testing or independent validation. So essentially lot of applications became customer focused. The risk of failure was extremely high.
	Making Assumptions	Refers to the judgements made for test cases' selection based on some assumptions that the Vendor team makes		
		Clients Thought Process: Indicates assumptions made about what client's thought processes might be behind some decisions or actions.	followup interview after observation	[S3T2.PM1] (Vendor Manager): But still we had shown the willingness that give us access to the XX software. But still they did not. Maybe logistics wise it must not have been possible. So, I don't know physically where that machine is and where our people are. So all those issues might also be there.
		Team's Configurations: Indicates that the global teams make assumptions about the roles and responsibilities of various members.	Interview with the Mexican Onsite Vendor Team Member	[S3.OC1]: I am on the understanding that PUS is the coach for the whole team over there in India. Right? (asking the researcher)
		Questioning Clients is Forbidden: Instances indicating assumptions made by vendors that questioning about business decisions is forbidden.	First Interview with participant post observation	[S3T2.TM2] (Vendor Manager): It's the client business. Everything we see is planned business. I will give you one example about the Client. They sell Cheese only in the North America. They will sell anywhere. It's the way they do. Ok then now we cannot ask the question- why you don't sell the cheese in Mexico and Latin America...They say "ok now this is not in the contract requirement, this is not the requirement" that's it. Then we will not be writing our testing and process.

Code System		Code Description	Data Collection Procedure	Sample Quote
		Requirements Documents are Thorough: Instances indicating that testers made assumptions based on what was and was not stated in the requirements documents.	Unstructured (ethnographic style) Interview with participant	[S3T3.TM1](Vendor Member): I'm not going to deliver that is anything beyond what is written in the requirements documents. But from what I have seen still the reaction from the onsite (client) site is that this is something that you should have anticipated. You should have asked me; I always thought that you were going to do this. And then the response from our side is that why didn't you tell me when we were taking sign off from you and you knew at that stage.
	Systematic Testing Standard	Providing a systematic and organizational standard for defect detection and test case creation procedure.	Group interview post client-vendor meeting observation	[S3.CO1]: So the reason we purposely didn't do that was because, the business tends…if they (client's business team) were going to do it they would miss steps, they don't have the time, they don't have a testing background to know what is a quality script. And then we lose the whole quality piece. (Vendor Team) just does a much better job at it (creating standardized high quality scripts)....
Clashing Planning Stategies		This demostrates the example of the clashing planning strategies.	followup interview after observation	[S3.DM1] (Vendor Manager) - We wanted to have this communication through to the KR-MX that first week "let's load less and let's pick up next week onwards." However, those guys were adamant that pick up as much as possible in the first week and then slowly slowly reduce it. So they picked up almost 60% of the target in the first week.
Cultural Facets		Instances illustrating cultural interplays		
Comparing/Describing Cultural Traits		Discussions in which participants describe or compare different cultures, their styles (of working), and thinking patterns. □ □	Unstructured (ethnographic style) Interview with participant	[S3T3.PM1] (Vendor Member): For the Americans when they show a mistake they actually show a mistake in the work and not in you. Where as for Japanese and for Indians when a mistake is shown, they do seem to take it personally. Also the Japanese tend to show or express mistakes with little sugar-coating so they do not try to be direct. But the Europeans and the Americans are much more direct.
Cross Cultural Influences (Acculturation)		Evidence of influence that one culture has on another in a Global Software Development Setting, which results in acculturation and adaptation of one culture into the other.	Unstructured (ethnographic style) Interview with participant	[HS] - so what is your opinion about this why-why analysis (It is an analysis method used by the Japan client team to investigate the root causes when many bugs emerge in testing process) [S3T3.TM1] (vendor member) - my opinion is that it is pretty much an eye opening experience once we also start thinking that much. Although it is quite painful, but then also it does two things (1) it brings us also closer to the root cause; it could be a root cause because of something that FUJ (Japan Client) did not do… that you didn't give me this. (2) it brings in a sense of discipline that this is what we did the last time and this is the pain that we had to go through (laughs subtly) to convince them (clients) about the each level of why, so next time onwards we will be more disciplined. Learn from the analysis that we did last time, the more disciplined, and try to adhere to the issues and targets at each level of testing. Stop testing if you're not reaching those levels. Find out why our target is exceeding. Because there are multiple levels of testing. One is which the developer does, one is unit testing, one is which the team will do coding and system testing. So

Code System	Code Description	Data Collection Procedure	Sample Quote
Cultural Facets Imposing Challenges	Cultural trends directly identified as challenges by the participants or indirectly interpreted as challenges by the researcher (i.e., indirectly deduced to be a challenge by the participant but not explicitly mentioned).	followup interview after observation	[S3T2.PM2] (Vendor Member): Yes, yes. We have not done a good job in terms of highlighting the risk content, pushing back to customer, since this is the risk if we are not convinced then this should not be done that way, we see that is a risk. If that is the risk we should have escalated big time that this is not going to work. We always have the normal Indian mentality like you know, this we will manage somehow, that is what they need we will try to deliver it. [Saying yes to clients requests.]
Cultural Knowledge Embedded in Test Cases	Evidence showing how cultural knowledge gets captured in the test cases and is embedded in the process of creating the test cases (onsite creates, explains to the offshore team, who then take it from there).	First Interview with participant post observation	[S3T2.TM1] (Vendor Member): Once we get to know what companies currently doing and how it is being implemented, we study that and then this gives us an idea. So ok, this is what I want to check out or this is what I want to test. So then depending upon that we go for Test Planning perspective. Then we design our Test Cases depending upon the business or the requirements whichever you call it. And then comes your part of execution and all other there are different matrix and all that we have.
Cultural Trait Being Enacted	Examples of cultural traits (such as power distance and the popular "always say yes" culture practice of Indians) being enacted through speech, action, or gestures.	followup interview after observation	[S3T2.PM2] (Vendor Team): We tried to do that curve projection but the customer pushed for this flat projection. They said "no we need everything" and we are not able to convince them. We were not able to pushback on them saying that this is not the right way of doing it and that is where the problem was. There with my onsite team because of the different culture because they were new, they were all Mexican folks, they were issues in terms of onsite-offsite communication. Leads were not able to convince the customer that this was not the right way of doing, so that was one of the reason that we over committed in the first two weeks and couldn't deliver, definitely it's not right. That was only definitely the big issue.
Culturally Sensitizing/Sensitized Clients	Expression of the idea of sensitizing the clients of the cultural differences or instances that demonstrate that the clients are culturally sensitive.	Group interview post client-vendor meeting observation	[S3.CO1]: then I've been pretty lucky, like the ladies they let me ask questions to learn about the culture. You know they don't take any visible offence. Maybe when I leave they do but… No but they are always willing to share [S3.DM1] (Vendor Member): this is what I was telling her it's not the first time a lady has been my friend.. There have been many but nobody asked the ladies to come in and ask the questions about how their life is, what they do once they got home so as to understand their way of working. Nobody has done that, so that's what I was telling her. She [LE] made that effort and that's why the closeness is there. There is no gap between the way she understands the Indians and the way Indians understand the Americans. But not everybody does that.

Code System		Code Description	Data Collection Procedure	Sample Quote
Culture Related Learning		Describes how knowledge of other cultures is acquired, what it consists of, and what is done with the knowledge acquired. Cultural knowledge is acquired in the form of language learning, learning their way of working etc. They put in effort in learning about another culture in subtle ways. □	followup interview after observation	[S3T2.PM2] (Vendor Member): I mean at my level I have not seen it much. We are taking the steps in the sense we are doing the workshops to explain them (Mexico Vendor Team) the Global Delivery Model (GDM), we are training them, we have a cultural program like what we do, you might have seen the cultural program for Mexico specifically where people working in Mexico attends. [HS]: Has these people attended it? [S3T2.PM2]: I am not sure, I have to check that. Similarly they have programs where in Mexico they have programs like what is Indian Culture. They have the programs there. So we are doing all that kinds of steps "yes".
	Acquire knowledge through failed experiences	Patterns where the teams fail to understand the other team's expectations at the beginning but later learn about their expectations through failed experiences.	followup interview after observation	[HS]: In Week 3? And there you didn't have to ask permission for? [S3T2.TM1] (Vendor Team): No, No because I have told them, you saw that, what happened in Week 2. So they allowed us to do the testcases from Week 4, in Week 3.
	Clients acquire it by interacting	Provides evidence of how clients acquire cultural learning of the vendor teams by interacting and asking them about their routine life.	Client Visit Observation	One of the client team members then asks … "Are you the one who got married to the Mexican team member?" They joke about one of the [USA] team member getting married to one of the Mexican team members….[This particular instance shows that the client team members are aware and updated about some personal level things going on at the vendor teams side.]
Culture's Influence on Business		Processes, practices, activities in business that vary because of different cultures and geographies. □	First Interview with participant post observation	[HS]: Do you think that even if the company is the same, the businesses in the different countries will be different? [S3T2.TM1]: Differs, yes. [HS]: So that here like for your level of work also there are differences in the way, things are done, is it? [S3T2.TM1]: Yes – Yes. See SAP implementation, if it would have been same. You don't need a different implementation, right? [HS]: That was the thing I was trying to understand, that you know where exactly the difference happen and why it is that? [S3T2.TM1]: Difference happens in many things. See if you say from technical perspective if you say, XX part, you have different currencies, Ok. You have different taxations; you can't map each and everything in that. Then, there are different products which get manufactured or which are marketed in that particular country. Ok. So these are major factors which affect. If you say Brazil. Brazil has a different kind of Taxation: they have 4-5 Taxation Code. US you say, entire country – one Taxation. Canada – it differs. Mexico – they have small Caricum countries. So now my business would say we have major sales here because manufacturing is

Code System		Code Description	Data Collection Procedure	Sample Quote
Culture's Influence on Testing		Evidences that exemplify and describe various ways in which culture influences testing as a software development activity.	followup interview after observation	[S3T2.PM2] (Vendor Team): Ok, for (test) planning...we always plan10-15% in the first week, 20% in the second week, slowly gradually increase, so by that time the environment stabilize the data issue is resolved, so we get the speed. It should be a curve; it cannot be a flat projection. We tried to do that curve projection but the customer pushed for this flat projection. They said "no we need everything" and we are not able to convince them. We were not able to pushback on them saying that this is not the right way of doing it and that is where the problem was.
	Testing scope and customizatio n	Evidence suggesting that testing scope varied geographically despite being the same client and the same software application being considered	First Interview with participant post observation	[HS]: Pre – Production is not a part of [Vendor V3] work? [S3T2.TM1] (Vendor Member): No. Not right now. In Mexico, we don't have, in some of the cases we do testing in Pre – Production also. [HS]: In Pre – Production also. Ok. [S3T2.TM1]: In Brazil, we did it for this.
	Influence of Test Case Distribution	Instances where test case distributions vary geographically, implying culture playing an underlying role in the test-case distribution decision-making process.	Online Meeting Observation	[S3.CO1] (USA Client Manager): OK (with frustration in the voice). I'm just trying to understand the concept…we are doing everything (i.e., all the testing) and they (MX-client team) are not. They are just doing it in the training tree. So just trying to understand how the concept is working really…
	Influence on Testing Strategy	Examples where testing strategies (e.g., test planning, test case creation) alter because of cultural influence.	followup interview after observation	[S3T2.PM1] (Vendor Team): Now why LE-Client (i.e., USA Client Manager) was raising concerns is that she was thinking that in cycle 1 some new functionality came, so process team (Mexico client-side team) tested that and we tested the existing functionality. Again in cycle two some additional functionality or some additional interfaces came, so process team tested that and we tested only the remaining scope in a special folder. Now she is thinking that in cycle three there should not be much new functionality. So ideally only we should be testing everything, process team should not be testing anything in the official folder. But still if you see the process team is saying that they will do these testing…so she is asking why then [Vendor V3] is not doing the entire testing, why process team is doing half the testing we are doing half the testing.
Flexibility Model		Cultural pattern of adaptability that is strongly emerging from the Indian team members' data.		
	Exploiting the Flexibilty Model	Evidence showing that misuse of the adaptability and flexibilty cultural facet results in exploitation of this facet by management and various teams/client members.	First Interview with participant post observation	[S3T2.TM1] (Vendor Team): But still we are again like doing nightouts or come on Saturday to work because client is going to release some of plans over on Friday. So we had a thing like, I have a Team of 4, for 2 team members I told them to take off on Monday, so that they have a long weekend and they can enjoy.
	Flexible or Rigid	Situations that exhibit the accomodative nature of the Indian team towards the onsite/client non-indian members or the accomodative/rigid nature of the client teams and displays the effort put by the Indian teams to work around the constraints of the non-indian teams.	followup interview after observation	[S3T2.TM2] (Vendor Member): I mean it was not easy for them to take a call in late nights or this time, so we get flexible too and ask them "Ok either this time works with you or this time works with you" We provide them with the flexibility in the call time.

Code System		Code Description	Data Collection Procedure	Sample Quote
	MX: Being Rigid about the Working Way	Inflexibilty towards altering the plans when things don't work as planned..	First Interview with participant post observation	[S3T2.TM1] (Vendor Member): And then it is like if "Product Costing" would have been done easily, we would have taken up on our own which Test Cases are there for Week 2 but that didn't happened at once. It has been released also but client was not allowing us to take it up from Week 2.
	Ven: Proposing an Alternate Plan	Instance where the offshore team shows flexibility by proposing an alternate plan.	followup interview after observation	[S3T2.PM1] (Vendor Member): But they (Mexico onsite-vendor team) don't get any real execution done there because they don't have the bandwidth (i.e. the time). But still we had shown the willingness that give us access to the XX software and they (Mexico onsite-vendor team) will do something. But still they (Mexico Client Team) did not.
	Ven: Relying on Flexibility Model	Evidence demonstrating the offshore team's reliance on their flexibility cultural model (Flexibility may be defined as the degree to which one culture adapts and adopts other culture and situations).	First Interview with participant post observation	[S3T2.TM1] (Vendor Member): So in Week 2 there was a bit like this in Cycle 1. And then we finished up, we were there for Saturday. We have completed and we have Week 2 Saturday (to work).
Language Challenges		Indication of language imposing a barrier for their smooth interaction with their MX counterparts.☐	First Interview with participant post observation	[S3T2.TM2] (Vendor Lead): They are more comfortable in Spanish and even all this e-mails and all, they are in Spanish language most of the times. So these people just give us (the information) "Okay this in English means like that." So they send it to us. Few times then if you want to really understand what is that we use the Google translator.
Saying No		Popular cultural pattern of "always saying yes" or hesitating to say no		
	Not Able To Say No	Instances where the participants admit that they are not able to say no to the clients.	followup interview after observation	[S3T2.PM2] (Vendor Manager): Everytime, we are not putting the foot down and saying I am the expert, this is the way only it should happen, we will not be able to do this. We are not good in doing that part. We are still "Ok" in communicating, we do communicate hierarchy issues. But I am very sure that if they do not take your big issue, it's going to create a failure for not only me but for the Client too. Actually we do not have the guts to go out and put our foot down and stop them in that minute of time. That is where we are weak.
	Essentials For Saying No	Information, reasoning, evidence that is required for a team member to say No to the request received from the clients.	followup interview after observation	[S3.DM1] (Vendor Manager): There is business worth 20 people that [onsite coordinator] wants us to do to get more credibility. There is delivery that is saying that this is a risk preposition we should not take it. Now I am in a situation where I have to come up with the best of the possible solutions. So I asked these guys (delivery team) have you done your capacity modeling?" They say "no." "if you have not been your capacity model then I don't know whether you can do it on not"
	Vendor Organizations Values System	Evidence suggesting that the busines-conduct ethical standards of the organization mentions about when teams should agree	Artifacts, Documents, and Chats Information	[Organization's Values Website]: "For the success of our client organizations we may need t o offer advice that is contradictory to what the client want s to hear"
Selective Information Hiding/Sharing		Evidence illustrating the offshore team's practice of selective sharing of information to avoid admitting any mistakes.	Group discussion over lunch	[S3.DM1] (Vendor Manager): Even in the Mexico project incident (referring to the escalations that occured), we said that we should inform about this to the client (LE-client). But VJ (onsite vendor manager) thought that it was not necessary to inform her. In fact I think that this was one of the best incidents we could have shared with them.

Code System	Code Description	Data Collection Procedure	Sample Quote
Developing a Consultancy Culture	Mention of the existence of a consultant's culture or expression of the need to build a consultant's culture in the organization. This culture is what the organization is currently aspiring for but its different from the delivery model culture that the organization is habituated to.		
Focus on What not Why	Patterns in the participants actions, words, behavior, attitudes that imply that their focus while providing the testing service to the clients has been on understanding "what is required" and not necessarily "why something is required."	First Interview with participant post observation	[S3T2.TM2] (Vendor Manager): It's the client business. Everything we see is planned business. I will give you one example about the Client. They sell Cheese only in the North America. They will sell anywhere. It's the way they do. Ok then now we cannot ask the question- why you don't sell the cheese in Mexico and Latin America...They say "ok now this is not in the contract requirement, this is not the requirement" that's it. Then we will not be writing our testing and process.
Failed at Consultancy Role	Evidence suggesting that the participants were failing at performing their consultant's role efficiently and effectively. (This also indirectly highlights the attributes that a consultant should possess).	followup interview after observation	[S3T2.PM2] (Vendor Manager): Everytime, we are not putting the foot down and saying I am the expert, this is the way only it should happen, we will not be able to do this. We are not good in doing that part. We are still "Ok" in communicating, we do communicate hierarchy issues. But I am very sure that if they do not take your big issue, it's going to create a failure for not only me but for the Client too. Actually we do not have the guts to go out and put our foot down and stop them in that minute of time. That is where we are weak.
Enacting as Consultants	Instances that depict the idea, which the organization was promoting, of they not only being the service providers to their clients but also consultants to them. They were thinking of giving the clients some suggestions on testing some modules not covered by the clients.	followup interview after observation	[S3.DM1] (Vendor Manager): I understand that concept. And then when he comes up with this problem I relate ikt with the multiple knowledges that I have gained. And then I have to advise him that this is not the way to do it; this is the right way to do it. This is what you have to focus on, don't hurry up. That is how you become a trusted advisor do I or do I not?
Expectations	How expectations are conveyed, interpreted by various team members.		
Interpretation of Offshore Team's Expectations	This code discusses the expectations that the offshore team has from the onsite coordinators and/or the client teams.	followup questions via email sent after having a semi-structured interview over the phone	[Followup question via email] What does the offshore team in India expect from you as an onsite coordinator? [S3.OC1] (Onsite Mexican Vendor Member): To have all the "what to do next", "how to solve the issues" and "what is the client thinking" answers for them on a daily basis.
Interpretation of Client's Expectation	Illustrates discussions that shed light on the client's expectations and the various team members (onsite team, offshore team's) understanding of those expectations.		
USA Clients Interpretation of MX Clients Expectations	Describes how USA client manager interprets MX client's expectations.	Online Meeting Observation	[S3.CO1] (Client Manager): It is critical then they [MX-client] needs to send us the information otherwise it is not critical. [sarcastic chuckle]

Code System		Code Description	Data Collection Procedure	Sample Quote
	Onsite Coordinators Interpretation of MX's Expectations	Describes the understanding that the onsite coordinator has of the Mexican client's expectations.	Interview with the Mexican Onsite Vendor Team Member	[S3.OC1] (Onsite Mexican Vendor Member): Visibility mean that the client wants to see the exactly what is [Vendor Team] doing in just a few words. I don't know maybe with colors, with excel sheets, with other point anything. They love visibility. The client doesn't want me to tell "ok I did this, I did this" and stuff like that. No, no, no. I want the visibility but per where are you going to.
	Vendor Interpretation of MX's Expectations	This is how the vendor team interprets the client expectation. They understand that whatever is on the contract has to be completed and tested. However, they offer to help in areas not on the contract, an effort to enhance the relationship with the clients.	Online Meeting Observation	[Meeting Attendee] (Onsite Mexican Vendor Member): At this point I am cautiously optimistic after Mexico team because it seems that after I send those couple of emails, there is finally some traction [I think he meant tension] and people are pushing to get this information to us asap.
GSD Description (Global Software Delivery Practice)		Describes characteristics of GSD, including terminologies used, standards followed, and concepts defined. It also includes what various teams mean in the GSD context, how the decision of outsourcing are done, what is the communication structure, how organizations conceptualize GSD differently, and what are the unsaid assumptions of the GSD model.		
Curve Projection Test Planning Model		Refers to conversations around the definition or discussion of the Curve projection model (which implies gradually increasing the test cases per week instead of a flat 25% per week plan).	followup interview after observation	[S3T2.PM2] (Vendor Team): Ok, for (test) planning...we always plan10-15% in the first week, 20% in the second week, slowly gradually increase, so by that time the environment stabilize the data issue is resolved, so we get the speed. It should be a curve; it cannot be a flat projection. We tried to do that curve projection but the customer pushed for this flat projection.
Building and Maintaining Relationship		Effort put in by the vendor teams to build and maintain good client relationship (e.g., build good rapport). Examples include additional services provided, requests processed, and procedures adhered as well as being flexible timewise, giving all the required information proactively (caring), and making small talks by asking about weekend activities (or weather related).	followup interview after observation	[S3T2.PM2] (Vendor Manager): We have not done a good job in terms of highlighting the risk content, pushing back to customer, since this is the risk if they are not convinced that this should not be done that ways, we see that is a risk. If that is the risk they should have escalated big time that this is not going to work. We always normal Indian mentality like you know, this we will manage somehow that is what they need we will try to deliver it and eventually we did it, if you see that escalation was there only for 1-2 days,right. Within a weekend, we asked the team to work on weekends, two days we work and on Monday when we back we were back on track
First Time Failing is Common		GSD cultural facet of failing to meet the deliverable for the first time is a common pattern observed in GSD culture.	First Interview with participant post observation	[S3T2.TM1](Vendor Member): In the starting it actually happens in this way. Because initially they are also new. Because this is an implementation, some people they are also totally new to the SAP. They also might not understand how the things happen. So it (failure) might happen
Fresh Graduates On The Team		Property of GSD cultural model of having a set of fresh graduate employees (inexperienced) on the team.	followup interview after observation	[S3T2.PM2] (Vendor Manager): See there were issues. tTere were issues 10-20% in some of the things. See everybody you will not, whatever we show on the presentation we should as if everybody is an expert, right. Out of 100 people 40 % are expert, 60% are fresher/trainee guys so they are going to do mistakes. But it is our model, we deliver it. We can't help it.

Code System	Code Description	Data Collection Procedure	Sample Quote
Get Maximum Work Offshore	Another characteristic of GSD is to get maximum work offshore.	followup interview after observation	[HS]: So then what motivates them (Onsite coordinators) to push so much when they realize that certain things cannot be done at offshore? [S3T2.PM2] (Vendor Manager): See they had two reasons to get motivated because this is our model. We have to get the maximum work done on offshore and second reason is if they want to retain the work they had, naturally they will end up doing everything.
GSD Culture Unawareness	Describes instances when participants express their or other team members unawareness of GSD culture	followup interview after observation	[S3.DM1] (Vendor Manager): We are used to the global delivery model, we are used to our people being deputed to work in different countries to work there, understand our expectations, working at night, and getting the things done. But when we recruit people in Mexico they are guided by their laws of the land. They can easily say that eight hours I am billing so I cannot work, I am not going to work more than that. So all those things initially bought up a lot of hiccup for us.
Team Structure and Configuration	Describes/Confirms the typical GDM structure where there is an offshore team, clients, and onsite coordinators that are typically Indians.	Informal followup discussion while walking back after weekly meeting observation	[S3T2.PM1]: In a typical GDP model they would have sent Indian folks on-site. But in this case, we have recruited the Mexican folks
Test Cases Detailing Process	Evidence illlustrating the main task of detailing the steps involved in the test cases creation process that the vendor-side test engineers usually perform.	Field notes from participant observation of a test engineer performing the task of creating test cases.	[Field Notes] Currently, the test engineer is writing steps in detail so that if a new person comes, he also knows how to execute the steps.
Working Extended Hours	Discussion of how working extended hours is a part of the routine for the team, they are reluctant to work for late hours, or they are force to work for late hours.	followup interview after observation	[S3.DM1] (Vendor Manager): We knew it could be done. [04:08] We worked over the weekends, we worked a little bit hard. And then we were able to reach the 80% adding four more days into it.
Intended vs Actual practice	Instances when the intended (formally agreed and expected) practice and actual (informally enacted) practice differ. Examples include helping outside contractual agreements (such as the documents, contracts) as well as diverging from the expected norm because of practical challenges(e.g., putting inexperienced testers because of shortage of experienced members). ☐	followup interview after observation	[HS]: You told me is that in cycle one most of the functionality is tested, that the bug fixes happen and then the cycle two. But over here now it looks like there is a different scenario. They are adding new functionality in cycle three? [S3T2.PM1] (Vendor Manager): Such things happen in any testing project. So the idea is that we do 3 cycles. They try to make maximum functionality available in the first cycle. But rarely it happens in any project that 100% functionality will be available in the first cycle. Although that is the target, many a times some development gets slipped and all. So, by the time cycle one starts, only 80% of the functionalities available. So we do that – 80% testing. Now the remaining 20% should be available in cycle two but sometimes it may happen that in cycle two also the development team may not have finished the remaining 20%.
Perception of Testing (IVS)	Description of the perception the various clients and vendor members hold towards testing as a practice.		

Code System	Code Description	Data Collection Procedure	Sample Quote
Business Meaning and Value of Testing	Vendor members' perception of the client's business (embodied in business objectives) meaning of software testing.□	Group discussion over lunch	[S3T2.PM2](Vendor Manager): The result of cycle one testing will decide the day the clients will be announcing the split of the organization in October. So, RD's (Client Manager's) thinking is that "I don't care about understanding the capability of the [Vendor team]. That is whether they can perform hundred percent testing, whether they are very productive etc. All I care about is to see whether the basic functionality in the splitted system works or not." So for us scoping means understanding which of the activities, test cases, have to be completed in the cycle one that will ensure that the basic functionality is working, and how should these things be planned.
Business Meaning of Bugs/Defects	Discussion around processes, metrics, and practices that indicate how the number of defects/bugs translate into business value (e.g., monetary loss), which inturn is interpretated in terms of the level of quality they can be delivered by an organization.	Focus group discussion with the Market Research Team member and a vendor manager to understand testing service from the market research point of view	[GU&Vsh] (Marketing Team): So you count the number of defects that come in one month or the three months that the system goes into production and then you will measure the defects that you have found during testing also. And then you figure out what's the quality that you have delivered.
Testing for Training	Some clients view testing for training purpose rather tha for verification and validation purpose.	Group interview post client-vendor meeting observation	[S3.CO1] (USA Client Manager): the biggest concern I hear from my clients is around the business as far as they go. Some of them use the testing to do training. So they're afraid if I use a testing team they are going to lose that opportunity.
Testing as Second Class Citizen	Confirms previous finding of viewing testing as a boring activity, metaphorically compared to being a second class citizen.	Focus group discussion with the Market Research Team member and a vendor manager to understand testing service from the market research point of view	[GU] (Vendor Manager): There are some challenges that are very very specific to testing alone because they decide to develop something because there is a need for that particular application to be developed. So no one asked a question as to why it has to be developed. Because when they start on and they decide to develop they would have done all the research, analysis as to what kind of returns they are getting. But the question that they asked is why do I need to test. If the application is coming then why do I really need to test.

Code System	Code Description	Data Collection Procedure	Sample Quote
Changing Perception of Testing (IVS)	Instances that demonstrate perceptions that have changed about the testing services and what are causing these changes.	Focus group discussion with the Market Research Team member and a vendor manager to understand testing service from the market research point of view	[S3.CMG1] (Market Researcher): Nowadays we see that in general testing as a word is being replaced. It is no longer being used. It is being replaced by quality assurance or validation. Our testing has become a kind of a subset of a much bigger function. Because now true validation is what our customers want is where you can align the validation to the business growth of an organization. That really is the end state that the client wants [09:24] and previous vendors, service providers in the ecosystem are responding in a different way.
Unproductive Productivity	Experience of performing an activity that is considered productive from one angle but it is unproductive from another angle. An activity that produces outcome but not value.	Email followup with participant	[S3T2.TM3] (Test Engineer): (Response to the email in which I asked followup questions) 1. Yes for every step in test case we need to attached snapshot (of the testcase step executed). 2. Attaching screenshot is mandatory. It is time consuming but what we are exactly doing [others] will come to know by seeing screenshots. 3. Simultaneously, we execute test case and attach screenshots. If we face any issue in qc then we can create one doc of all screens and then upload stepwise in Quality Center.
Trust	Describes instances when the client might lose trust in the vendor organization or actions that vendor organizations take to gain/regain trust of the vendor organizations.	Artifacts, Documents, and Chats Information	[From the Annual Feedback Form Provided By Client To Vendor]: "Estimates should include contingency time? Don't be overly aggressive. Be realistic. This includes small deliverables not just major milestones. Credibility is extremely important."
Damage Control to Regain Client's Trust	Actions taken to regain the trust of their clients again	followup interview after observation	[S3T2.PM2] (Vendor Manager): I get it done within the Client team itself, not from my side but from the other side. So I arranged for a few people myself, who are expert like there were people from XX's team, people from YY's team. So I said that, "You look at it from an independent angle." One is independent angle and second is because they were not having time to do the audit, they were busy in execution, so I got the team from within [KR] but outside Mexico for 2-3 days and they did the audit. We had listed what all are the issues there and what we corrected.

Code System	Code Description	Data Collection Procedure	Sample Quote
Lost Client's Confidence	Description of how participants ended up losing the client's confidence.	followup interview after observation	[S3T2.PM2]: the reason for that we didn't plan properly, we lost one whole week and what happens in that when one something goes wrong, what will happen, we lost the confidence or we will lose the confidence. What will they do? They will like to check everything, right? Now they started checking everything---whether these people are doing what they are saying. They got a doubt that either they are doing some cheating or they are just...You know how come if you are not able to do it (execute the test cases) in 6 six days and now in 2-3 days you are able to do it. So I am a human mind, I am going to doubt what you are doing. So what they did they started doing the audit "I want to check what execution you have done till now." Unfortunately, we got one or two issues. See it happens---your 80% testing is accurate but 20% there will be issues. Issues in the sense we have not done cheating but we have done fast runs. For example, fast run in the sense that without recording the proof of testing, I have just passed the test case.
Reluctance To Trust Offshore Testing	Evidence that shows that the client team was reluctant to trust the offshore testing setting.	Client Visit Observation	[S3.CO1]: Yes, you said that any new functionality has to be tested by [[KR] Client] first and after that eventually [Vendor Team] can do the testing.
Screenshots as Proof Of Test	Evidence suggesting that screenshots was the way for clients to trust that offshore teams actually performed their testing activites.	followup interview after observation	[S3T2.PM2] (Vendor Manager): When you say that the test case I executed it, the status saying passed, right, but at the same time I am suppose to attach the screen shot, write on the details that this is what I have done and this is the proof of doing that.
Understanding Reasons for Escalation, Failure	Describes the scenario when the offshore team was very close to losing the MX client and its project as well as the description of different participants' perspectives on the reasons for failure that led to escalations in the MX team.	followup interview after observation	[S3.DM1]: That's where the communication gap was very problematic. The said yes and we said okay [he emphasized the word "okay"] rather than saying that not okay. And this is what exactly happened. By the end of the first week the we were nowhere near the 60% time (in terms of completing the testing cases). And that's what the escalation was all about ..."[Vendor V3] does not have the capability, they do not have the communication skills, they don't have the attitude"
Vendors' Perception of Clients Roles and Responsibilities	This provides instances of discussion when the vendor participant reveal their understanding of the client's team structures, roles, and responsibilities.	followup interview after observation	[S3T2.TM2]: ...she is the client contact for us...and we keep her updated on regular basis like we did today in this meeting ...it's her responsibility to manage the vendor team. [HS]: Oh! Ok. So it's her responsibility with respect to all the different regions? [S3T2.TM2]: ... she is responsible because we put her name as NRI reporting vendor. She makes sure that we have proper contract documents signed with this vendor only. What are the activities and all that? So her job is to make sure Vendor performs whatever is been expected for her contract.

APPENDIX F

IRB APPROVAL

CONSENT DOCUMENT FOR ENROLLING ADULT PARTICIPANTS IN A RESEARCH STUDY

Georgia Institute of Technology

Project Title: "Qualitative Study to Understand the Influence of Culture on Global Software-Testing Practice"

Investigators: Mary Jean Harrold, ADVANCE Professor/College of Computing (Faculty Co-Advisor) Nancy Nersessian, Professor of Cognitive Science/School of Interactive Computing (Faculty Co-Advisor) Hina Shah, Graduate Student/College of Computing and School of Interactive Computing

Introduction/Purpose(s) of the Research:
This research study is to understand the human aspects involved in testing practice with an aim to identify some social and cultural issues that they encounter while performing their tasks. We expect that this project will provide some deeper insights into human aspects of software engineering problems that are mainly focused on software testing related tasks.

Research Funding:
The research study is currently not funded by any grants.

Procedure:
We request your participation in this study that focuses on understanding how testing is practiced in global software engineering (testing) setting in the industry. For the study, we will gather information in three ways – observations, interviews, and meetings. We will start with giving a brief introduction about the scope of our project to you; then we will ask you to participate in informal one-on-one meetings (e.g., to introduce each other, give an opportunity to the researcher to understand your role in the project, and in the organization as a whole); based on available opportunities, we will observe you do some of your activities (with your permission) and participate in short conversations during breaks or free times; finally, we will conduct an interview at the end to confirm our understanding of what we observed.

Foreseeable Risks or Discomforts:
There are no foreseeable risks or discomforts in this study.

Benefits:
There is no expected direct benefit for individual participants. However, through the study we hope to help the participants by (1) identifying some (individual, social, psychological, cultural, and/or organizational,) problems that software engineers commonly encounter while performing their tasks so that solutions can be designed for them, and (2) identifying some good trends and patterns of work practices that people adopt and share them with an aim to support people's tasks better.

Compensation/Costs:
We provide no compensation for participation and there will be no financial cost to the participants.

Consent Form approved by Georgia Tech IRB from 06 July, 2011 to Indefinite.

Confidentiality:
We will keep all information concerning you private and data collected about you to the extent allowed by law. We will not use your name and any other information that might identify you when results of this study are presented or published. Also, this information will not be shared with any other members in your company and it will not be used against you in any way. To ensure that this research is being conducted out in the proper way, the Georgia Institute of Technology IRB (Institutional Review Board) will review study records. The Office of Human Research Protections may also look over study records during required reviews.

Injury/Adverse Reactions:
Please report to Mary Jean Harrold (the faculty advisor and principal investigator for this project) at (404) 385-0612 for any injuries or reaction. Neither the Georgia Institute of Technology nor the principal investigator has made provisions for payment of costs associated with any injury resulting from participation in this study.

Contact Persons:
If you have questions about the research, call or write Mary Jean Harrold (Principal Investigator) at:

College of Computing
Georgia Institute of Technology
266 Ferst Drive, Atlanta, GA 30332-0765 Office: 2322 Klaus Advanced Computing Building
(404) 385-0612

Voluntary Participation/Withdrawal:
Your participation in this study is voluntary. You have rights as a research volunteer. There is no penalty for not participating. You have the right to change your mind and leave the study at any time without giving any reason and without penalty. Any new information that may make you change your mind about being in this study will be given to you. You may stop participating in this study at any time with no penalty. If you decide not to finish the study, you have the right to withdraw any data collected about you. If you withdraw, we will delete all your data in soft copies and shred the hardcopies. You do not waive any of your legal rights by participating in this research project.

If you have any questions about your rights as a research volunteer, call or write:
IRB Administrator
Office of Research Compliance
Georgia Institute of Technology, Atlanta, GA 30332-0420
Phone: (404) 894-6949

Signatures:
If you answer the following questions, it indicates that you have read the information contained in this letter, the researchers have answered all of your questions to your satisfaction, and you consent to volunteer for this study. You will be given a copy of this consent form to keep.

1) My designation at the organization is _____

2) I have _____(months/years) of experience as a _____

3) I have been working at the current organization since _____(month and year)

Thank you, (Signatures of Investigators)

Investigator's Signature: _____Date:_____

References

[1] "Improved cross-cultural communication increases global sourcing productivity." http://newsroom.accenture.com/article_display.cfm?article_id=4376, Jul 2006.

[2] "Execution is everything: The keys to offshore success." http://www.atkearney.com/images/global/pdf/offshoringfinal_s.pdf, 2007.

[3] "Overcome cultural differences in the outsourcing process." http://www.osf-global.com/, Apr 2009.

[4] "Top software testing and quality assurance outsourcing vendors." http://www.datamonitor.com/, 2009.

[5] Abraham, L. R., "Cultural differences in software engineering," in *Proc. ISEC*, pp. 95–100, 2009.

[6] Al-Ani, B., Bietz, M. J., Wang, Y., Trainer, E., Koehne, B., Marczak, S., Redmiles, D., and Prikladnicki, R., "Globally distributed system developers: Their trust expectations and processes," in *Conference on Computer Supported Cooperative Work (CSCW)*, pp. 563–574, Feb 2013.

[7] Al-Ani, B., Redmiles, D., de Souza, C. R., Prikladnicki, R., Marczak, S., Lanubile, F., and Calefato, F., "Trust in virtual teams: Theory and tools," in *Proc. of Conference on Computer Supported Cooperative Work*, (New York, NY, USA), pp. 301–306, ACM, 2013.

[8] Al-Ani, B., Trainer, E., Redmiles, D., and Simmons, E., "Trust and surprise in distributed teams: Towards an understanding of expectations and adaptations," in *Proc. of International Conference on Intercultural Collaboration*, pp. 97–106, Mar 2012.

[9] Ali Babar, M., Verner, J. M., and Nguyen, P. T., "Establishing and maintaining trust in software outsourcing relationships: An empirical investigation," *Journal of Software Systems*, vol. 80, pp. 1438–1449, September 2007.

[10] Aranda, J. and Easterbrook, S., "Distributed cognition in software engineering research: Can it be made to work?," in *Proc. CSCW Workshop on Supporting the Social Side of Large Scale Software Development*, Dec. 03 2006.

[11] Aranda, J. and Venolia, G., "The secret life of bugs: Going past the errors and omissions in software repositories," in *Proc. ICSE*, vol. 0, (Los Alamitos, CA, USA), pp. 298–308, IEEE Computer Society, 2009.

[12] Aspray, W., Mayadas, F., and Vardi, M. Y., "Globalization and Offshoring of Software: A Report of the ACM Job Migration Task Force." http://www.acm.org/globalizationreport/pdf/fullfinal.pdf, 2006.

[13] BAGRI, N. T., "After boeing 787 is diverted, air india looks into software problem." http://www.nytimes.com/2014/02/07/business/after-boeing-787-is-diverted-air-india-looks-into-software-problem.html?_r=0, Feb 2014.

[14] Begel, A., Khoo, Y. P., and Zimmermann, T., "Codebook: discovering and exploiting relationships in software repositories," in *Proc. ICSE*, pp. 125–134, 2010.

[15] Begel, A. and Simon, B., "Novice software developers, all over again," in *Proc. International Workshop on Computing Education Research*, (New York, NY, USA), pp. 3–14, ACM, 2008.

[16] Begel, A. and Simon, B., "Struggles of new college graduates in their first software development job," in *SIGCSE* (Dougherty, J. D., Rodger, S. H., Fitzgerald, S., and Guzdial, M., eds.), pp. 226–230, ACM, 2008.

[17] Beizer, B., *Software Testing Techniques*. 2 ed., 1990.

[18] Bentley, R., Hughes, J. A., Randall, D., Rodden, T., Sawyer, P., Shapiro, D., and Sommerville, I., "Ethnographically-informed systems design for air traffic control," in *Proc. CSCW*, (New York, NY, USA), pp. 123–129, ACM, 1992.

[19] Bird, C., Murphy, B., Nagappan, N., and Zimmermann, T., "Empirical software engineering at microsoft research," in *Proc. CSCW*, pp. 143–150, 2011.

[20] Bird, C., Nagappan, N., Devanbu, P. T., Gall, H., and Murphy, B., "Does distributed development affect software quality? an empirical case study of windows vista," in *Proc. ICSE*, pp. 518–528, 2009.

[21] Boden, A., Avram, G., Bannon, L., and Wulf, V., "Knowledge sharing practices and the impact of cultural factors: reflections on two case studies of offshoring in sme," *Journal of Software Maintenance and Evolution: Research and Practice*, 2010.

[22] Borchers, G., "The software engineering impacts of cultural factors on multi-cultural software development teams," in *Proc. ICSE*, pp. 540–545, 2003.

[23] Braun, V. and Clarke, V., "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, pp. 77–101, 2006.

[24] Breu, S., Premraj, R., Sillito, J., and Zimmermann, T., "Investigating information needs to improve cooperation between developers and bug reporters," in *Proc. CSCW*, February 2010.

[25] Brooks, F. J., *The Mythical Man-Month*. Addison Wesley, 1995.

[26] Calefato, F., Damian, D., and Lanubile, F., "An empirical investigation on text-based communication in distributed requirements workshops," in *Proc. ICGSE*, pp. 3–11, 2007.

[27] Camp II, R. D., *Effects of National Culture on Trust Development: A study of Canadian and Japanese Business Students*. PhD thesis, The University of British Columbia, 2002.

[28] Carter, T., "Cheaper's not always better." `http://www.drdobbs.com/architecture-and-design/184415486?_requestid=247829`, Mar 2006.

[29] Casado-Lumbreras, C., Colomo-Palacios, R., Soto-Acosta, P., and Misra, S., "Culture Dimensions in Software Development Industry: The Effects of Mentoring," *Scientific Research and Essays*, vol. 6, no. 11, pp. 2403–2412, 2011.

[30] Casey, V., "Leveraging or exploiting cultural difference?," in *Proc. ICGSE*, pp. 8–17, 2009.

[31] Casey, V., *Software Testing and Global Industry: Future Paradigms*. United Kingdom: Cambridge Scholars Publishing, 2009.

[32] Casey, V., "Imparting the importance of culture to global software development," *ACM Inroads*, vol. 1, pp. 51–57, September 2011.

[33] Casey, V. and Richardson, I., "Project Management within Virtual Software Teams," in *Proc. ICGSE*, 2006.

[34] Cataldo, M., Bass, M., Herbsleb, J. D., and Bass, L., "On coordination mechanisms in global software development," in *Proc. ICGSE*, ICGSE '07, pp. 71–80, 2007.

[35] Cataldo, M., Shelton, C., Choi, Y., Huang, Y.-Y., Ramesh, V., Saini, D., and Wang, L.-Y., "Camel: A tool for collaborative distributed software design," in *Proc. ICGSE*, pp. 83–92, 2009.

[36] Cater-Steel, A. and Toleman, M., "The impact of national culture on software engineering practices," *International Journal of Technology, Policy and Management*, vol. 8, no. 1, pp. 76–90, 2008.

[37] Cefkin, M., ed., *Ethnography and the Corporate Encounter: Reflections on Research In and Of Corporations.* New York: Berghahn Books, 2009.

[38] Conchuir, E. O., Holmstrom, H., Agerfalk, P. J., and Fitzgerald, B., "Exploring the assumed benefits of global software development," in *Proc. ICGSE*, pp. 159–168, 2006.

[39] Cramton, C. D. and Hinds, P. J., "An embedded model of cultural adaptation in global teams," *Organization Science*, no. 0, pp. 1–26, 2014.

[40] Cuff, D., *Architecture: The Story of Practice*. Architecture/Sociology, MIT Press, 1992.

[41] Curtis, B., Krasner, H., and Iscoe, N., "A field study of the software design process for large systems," *CACM*, vol. 31, no. 11, pp. 1268–1287, 1988.

[42] da Silva, F. Q. B., Costa, C., Franca, A. C. C., and Prikladinicki, R., "Challenges and solutions in distributed software development project management: A systematic literature review," in *Proc. ICGSE*, pp. 87–96, 2010.

[43] Damian, D., Izquierdo, L., Singer, J., and Kwan, I., "Awareness in the Wild: Why Communication Breakdowns Occur," in *Proc. ICGSE*, pp. 81–90, 2007.

[44] Damian, D. E. and Zowghi, D., "An insight into the interplay between culture, conflict and distance in globally distributed requirements negotiations," in *Proc. HICSS*, 2003.

[45] de Souza, C. R. B., *On the Relationship between Software Dependencies and Coordination: Field Studies and Tool Support*. PhD thesis, University of California, Irvine, 2005.

[46] Deshpande, S., Richardson, I., Casey, V., and Beecham, S., "Culture in global software development – a weakness or strength?," in *Proc. ICGSE*, pp. 67–76, 2010.

[47] Dimoka, A., "What does the brain tell us about trust and distrust? evidence from a functional neuroimaging study," *Management Information Systems Quarterly*, vol. 34, no. 2, pp. 373—396, 2010.

[48] Dittrich, Y. and Giuffrida, R., "Exploring the role of instant messaging in a global software development project," in *Proc. ICGSE*, 2011.

[49] Eaton, T., *A cultural analysis of information technology offshore outsourcing: an exercise in multi-sited ethnography of virtual work*. PhD thesis, Wayne State University, 2011.

[50] Fendler, J. and Winschiers-Theophilus, H., "Towards contextualised software engineering education: an african perspective," in *Proc. ICSE*, pp. 599–607, 2010.

[51] Geertz, C., *Thick Description: Toward an Interpretive Theory of Culture*. New York: Basic Books, 1973.

[52] Gertsen, M. C. and Sø derberg, A.-M., "Expatriate stories about cultural encounters: A narrative approach to cultural learning processes in multinational companies," *Scandinavian Journal of Management*, vol. 26, no. 3, pp. 248–257, 2010.

[53] Grechanik, M., Jones, J. A., Orso, A., and van der Hoek, A., "Bridging Gaps between Developers and Testers in Globally-Distributed Software Development," in *Proc. FSE/SDP Workshop on Future of Software Engineering Research*, FoSER '10, pp. 149–154, 2010.

[54] Grinter, R. E., *Understanding Dependencies: A Study of the Coordination Challenges in Software Development*. PhD thesis, University of California, Irvine, 1996.

[55] Guo, P. J., Zimmermann, T., Nagappan, N., and Murphy, B., ""Not my bug!" and other reasons for software bug report reassignments," in *Proc. CSCW*, pp. 395–404, 2011.

[56] Gupta, M. and Fernandez, J., "How globally distributed software teams can improve their collaboration effectiveness?," in *Proc. ICGSE*, 2011.

[57] Hahn, E. D. and Bunyaratavej, K., "Services Cultural Alignment in Offshoring: The Impact of Cultural Dimensions on Offshoring Location Choices.," *Journal of Operations Management*, vol. 28, no. 3, pp. 186–193, 2010.

[58] Hall, E. T., *The Silent Language*. Greenwood Press, 1959.

[59] Hall, E. T., *Beyond Culture*. New York: ANCHOR Books, 1976.

[60] Hayes, N., *Doing qualitative analysis in psychology*. Psychology Pr, 1997.

[61] Hazzan, O. and Dubinsky, Y., "Clashes between culture and software development methods: The case of the israeli hi-tech industry and extreme programming," in *Proc. Agile Development Conference*, pp. 59–69, 2005.

[62] Herbsleb, J. D., Paulish, D. J., and Bass, M., "Global software development at siemens: Experience from nine projects," in *Proc. ICSE*, pp. 524–533, 2005.

[63] Herbsleb, J. and Mockus, A., "An empirical study of speed and communication in globally-distributed software development - herbsleb," *Transactions on Software Engineering*, vol. 29, no. 6, pp. 481–494, 2003.

[64] Hestres, L., "The influence of american culture on software design: Microsoft outlook as a case study," *Gnovis: GeorgeTown University's Journal on Communication, Culture, and Technology*, vol. 3, 2003.

[65] Hinds, P., Liu, L., and Lyon, J., "Putting the global in global work : an intercultural lens on the practice of cross-national collaboration," *Annals / The Academy of Management*, vol. 5, pp. 135 – 188, 2011.

[66] Hinds, P. and Mortensen, M., "Understanding conflicts in geographically distributed teams: the moderating effects of shared identify, shared context, and spontaneous communication," *Organization Science*, vol. 16, pp. 290–307, 2005.

[67] Hofstede, G., *Culture's Consequences: Comparing Values, Behaviors, Institutions and Organizations Across Nations*. Thousand Oaks CA: Sage Publications, 2001.

[68] Hofstede, G. and Hofstede, G.-J., *Cultures and Organizations: Software of the Mind*. McGraw-Hill U.S.A.,, 3 ed., 2010.

[69] Holland, D. and Quinn, N., *Cultural Models in Language & Thought*. Cambridge University Press, 1992.

[70] Holmstrom, H., Conchuir, E. O., Agerfalk, P. J., and Fitzgerald, B., "Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance," in *Proc. ICGSE*, pp. 3–11, 2006.

[71] Hutchins, E., *Cognition in the Wild*, vol. 19. MIT press Cambridge, MA, 1995.

[72] Hutchins, E., "How a cockpit remembers its speed," 1995.

[73] Imsland, V., *The Role of Trust in Global Software Outsourcing Relationships*. PhD thesis, University of Oslo, 2003.

[74] Irani, L., Vertesi, J., Dourish, P., Philip, K., and Grinter, R. E., "Postcolonial computing: a lens on design and development," in *Proc. CHI*, pp. 1311–1320, 2010.

[75] Irani, L. C. and Dourish, P., "Postcolonial interculturality," in *Proc. IWIC*, p. 249, 2009.

[76] Jalali, S., Gencel, C., and Šmite, D., "Trust dynamics in global software engineering," in *Proc. of International Symposium on Empirical Software Engineering and Measurement*, ESEM '10, (New York, NY, USA), pp. 23:1–23:9, ACM, 2010.

[77] John A. Hughes, lan Somerville, R. B. and Randall, D., "Designing with ethnography: making work visible," *Sociology and Computing Departments, University of Lancaster, Bailrigg, Lancaster LA1 4YR, UK*, vol. 5, 1993.

[78] Jones, J. A., Grechanik, M., and van der Hoek, A., "Enabling and enhancing collaborations between software development organizations and independent test agencies," in *Proc. ICSE Workshop on Cooperative and Human Aspects on Software Engineering*, pp. 56–59, 2009.

[79] Jordan, B., ed., *"Advancing Ethnography in Corporate Environments: Challenges and Emerging Opportunities"*. Left Coast Press, 2012.

[80] Jordan, B. w. M. L., *Working in Corporate Jungles: Reflections on Ethnographic Praxis in Industry*, ch. Working in Corporate Jungles: Reflections on Ethnographic Praxis in Industry, pp. 95–133. New York: Berghahn Books, 2009.

[81] Ko, A. J., DeLine, R., and Venolia, G., "Information needs in collocated software development teams," in *Proc. ICSE*, pp. 344–353, 2007.

[82] Kobayashi-Hillary, M., "A passage to india," *Queue – Quality Assurance*, vol. 3, no. 1, pp. 54–60, 2005.

[83] Krishna, S., Sahay, S., and Walsham, G., "Managing cross-cultural issues in global software outsourcing," *CACM*, vol. 47, pp. 62–66, 2004.

[84] Kroeber, A. L. and Kluckhohn, C., *Culture: A critical review of concept and definition*. Kraus Reprint Co, 1978.

[85] Kvedaraviciene, G. and Boguslauskas, V., "Underestimated Importance of Cultural Differences in Outsourcing Arrangements," *Inzinerine Ekonomika-Engineering Economics*, vol. 21, no. 2, pp. 187–196, 2010.

[86] Laurent, P., Mäder, P., Cleland-Huang, J., and Steele, A., "A Taxonomy and Visual Notation for Modeling Globally Distributed Requirements Engineering Projects," in *Proc. ICGSE*, 2010.

[87] Levina, N. and Vaast, E., "Innovating or doing as told? status differences and overlapping boundaries in offshore collaboration," *Management Information Systems Quarterly*, vol. 32, pp. 307–332, June 2008.

[88] Lings, B., Lundell, B., Agerfalk, P. J., and Fitzgerald, B., "A Reference Model for Successful Distributed Development of Software SystemsSS," in *Proc. ICGSE*, ICGSE '07, pp. 130–139, 2007.

[89] Lytle, A., Brett, J., Barsness, Z., Tinsley, C., and Janssens, M., "A paradigm for confirmatory cross-cultural research in organizational-behavior," in *Research in organizational behavior: an annual series of analytical essays and critical reviews*, vol. 17, pp. 167–214, Jai press inc, 1995.

[90] Malinowski, B., *Argonauts of the Western Pacific*. Waveland Press, Inc, 1922.

[91] Manteli, C., Hooff, B. v. d., Tang, A., and Vliet, H. v., "The Impact of Multi-site Software Governance on Knowledge Management," in *Proc. ICGSE*, 2011.

[92] Markus, M. and Keil, M., "If we build it, they will come: Designing information systems that people want to use," *Sloan Management Review*, vol. 35, no. 4, pp. 11–25, 1994.

[93] Martin, D., Rooksby, J., Rouncefield, M., and Sommerville, I., "'good' organisational reasons for 'bad' software testing: An ethnographic study of testing in a small software company," in *Proc. ICSE*, (Washington, DC, USA), pp. 602–611, 2007.

[94] Mason, G., "The assumption bias and testing: How does it influence you?," in *Tea-Time with Testers*, 2013.

[95] McSweeney, B., "Hofstede's model of national cultural differences and their consequences: A triumph of faith - a failure of analysis," *Human Relations*, vol. 55, pp. 89–118, 2002.

[96] Millen, D. R., "Rapid ethnography: Time deepening strategies for hci field research," in *Proc. Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, pp. 280–286, 2000.

[97] Moe, N. B., Dingsøyr, T., and Dybå, T., "Understanding self-organizing teams in agile software development," in *Proc. Australian Conference in Software Engineering*, pp. 76Ű–85, 2008.

[98] Monasor, M., *A framework for interaction training in Global Software Development*. PhD thesis, University of Castilla-La Mancha, Ciudad Real, 2014.

[99] Morales, A. W., "Offshore by the numbers." `http://www.drdobbs.com/184415074`, Jan 2004.

[100] Myers, G. J., *The Art of Software Testing*. John Wiley & Sons, 1979.

[101] Niinimaki, T. and Lassenius, C., "Experiences of instant messaging in global software development projects: A multiple case study," in *Proc. ICGSE*, 2008.

[102] Olson, J. S. and Olson, G. M., "Culture surprises in remote software development teams," *Queue*, vol. 1, pp. 52–59, December 2003.

[103] Patton, M. Q., "Enhancing the quality and credibility of qualitative analysis," *Health Services Research*, vol. 34, no. 5, pp. 1189–1208, 1999.

[104] Philip, T., Schwabe, G., and Ewusi-Mensah, K., "Critical Issues of Offshore Software Development Project Failures," *ICIS*, p. 77, 2009.

[105] Pressman, R. S., *Software Engineering: A Practitioners Approach*. McGraw-Hill, sixth ed., 2005.

[106] Prior, J., *Everyday Practices of Agile Software Developers*. PhD thesis, University of Technology, Sydney, Jul 2011.

[107] Prior, J., Robertson, T., and Leaney, J., "Situated software development: Work practice and infrastructure are mutually constitutive," in *Proc. Australian Conference on Software Engineering*, (Washington, DC, USA), pp. 160–169, IEEE Computer Society, 2008.

[108] Ramasubbu, N., Cataldo, M., Balan, R. K., and Herbsleb, J., "Configuring Global Software Teams: A Multi-Company Analysis of Productivity, Quality, and Profits.," in *Proc. ICSE*, 2011.

[109] Robillard, M., Coelho, W., and Murphy, G., "How effective developers investigate source code: An exploratory study," *Transactions on Software Engineering*, vol. 30, no. 12, pp. 889–903, 2004.

[110] Robinson, H. and Sharp, H., "Organisational culture and xp: three case studies," in *Proc. Agile Conference*, 2005.

[111] Robinson, H., Segal, J., and Sharp, H., "Ethnographically-informed empirical studies of software practice," *Information and Software Technology*, vol. 49, no. 6, pp. 540–551, 2007.

[112] Rönkkö, K., ""yes - what does that mean?" understanding distributed requirements handling," in *Social Thinking - Software Practice*, pp. 223–241, 2002.

[113] Rooksby, J., Rouncefield, M., and Sommerville, I., "Testing in the wild: The social and organisational dimensions of real world practice," *Proc. CSCW*, vol. 18, no. 5-6, pp. 559–580, 2009.

[114] Rönkkö, K., *Making Methods Work in Software Engineering Method Deployment - as a Social Achievement*. PhD thesis, Blekinge Institute of Technology, Sweden, 2005.

[115] Segal, T. and Vasilache, S., "Plea against cultural stereotypes," in *Proc. ICIC*, pp. 275–278, 2010.

[116] Serce, F. C., Alpaslan, F.-N., Swigger, K., Brazile, R., Dafoulas, G., Lopez, V., and Schumacker, R., "Exploring collaboration patterns among global software development teams," in *Proc. ICGSE*, pp. 61–70, 2009.

[117] Shah, H. and Harrold, M. J., "Studying human and social aspects of testing in a service-based software company: case study," in *Proc. CHASE*, pp. 102–108, 2010.

[118] Shah, H., Harrold, M. J., Nersessian, N., and Newstetter, W., "Studying the influence of culture in global software engineering: Thinking in terms of cultural models," in *Proc. ICIC*, 2012.

[119] Shah, H., Harrold, M. J., and Sinha, S., "Global software testing under deadline pressure: Vendor-side experiences," *Information & Software Technology*, vol. 56, no. 1, pp. 6–19, 2014.

[120] Shah, H., Sinha, S., and Harrold, M. J., "Outsourced, offshored software-testing practice: Vendor-side experiences," in *Proc. ICGSE*, 2011.

[121] Sharp, H., Baddoo, N., Beecham, S., Hall, T., and Robinson, H., "Models of motivation in software engineering," *Information and Software Technology*, vol. 51, no. 1, pp. 219 – 233, 2009. Special Section - Most Cited Articles in 2002 and Regular Research Papers.

[122] Sharp, H., Hall, T., Baddoo, N., and Beecham, S., "Exploring motivational differences between software developers and project managers," in *Proc. ESEC-FSE*, pp. 501–504, 2007.

[123] Sharp, H. and Robinson, H., "An ethnographic study of XP practice," *Empirical Software Engineering*, vol. 9, no. 4, pp. 353–375, 2004.

[124] Shore, B., *Culture in mind: Cognition, culture, and the problem of meaning*. Oxford University Press, USA, 1996.

[125] Sillito, J., Murphy, G. C., and De Volder, K., "Questions programmers ask during software evolution tasks," in *Proc. Iinternational Symposium on Foundations of Software Engineering*, pp. 23–34, 2006.

[126] Sinha, V. S., Sengupta, B., and Ghosal, S., "An adaptive tool integration framework to enable coordination in distributed software development," in *Proc. ICGSE*, pp. 151–155, 2007.

[127] Sommerville, I., Rodden, T., Sawyer, P., and Bentley, R., "Sociologists can be surprisingly useful in interactive systems design," in *Proc. HCI Conference on People and Computers VII*, pp. 341–353, 1992.

[128] Sommerville, I. and Rouncefield, M., "Ethnography and software testing." `http://www.cs.st-andrews.ac.uk/files/EthnoAndTesting2.pdf`, Nov 2007.

[129] Spanjers, H., ter Huurne, M., Graaf, B., Lormans, M., Bendas, D., and van Solingen, R., "Tool support for distributed software engineering," in *Proc. ICGSE*, pp. 187–198, 2006.

[130] Spradley, J., *Participant Observation*. 1980.

[131] Spradley, J. P., *The Ethnographic Interview*. New York: Holt, Rinehart adn Winston, 1979.

[132] Strauss, A. and Corbin, J., *Basics of Qualitative Research*. Thousand Oaks, California: Sage Publications, 2 ed., 1998.

[133] Suchman, L., "Making work visible," *CACM*, vol. 38, no. 9, pp. 56–ff., 1995.

[134] Suchman, L. A., *Plans and situated actions: The problem of human-machine communication*. Cambridge U.K.: Cambridge University Press, 1987.

[135] Taipale, O. and Smolander, K., "Improving software testing by observing practice," in *Proc. International Symposium on Empirical Software Engineering*, (New York, NY, USA), pp. 262–271, ACM, 2006.

[136] Tassey, G., "The economic impacts of inadequate infrastructure for software testing," *National Institute of Standards and Technology, RTI Project*, 2002.

[137] Treitlier, I., "Backyard ethnography: defamiliarize the familiar to transform business (to appear)," *International Journal of Business Anthropology*, 2014.

[138] Treude, C. and Storey, M.-A., "How tagging helps bridge the gap between social and technical aspects in software development," in *Proc. ICSE*, pp. 12–22, 2009.

[139] Treude, C. and Storey, M.-A., "Awareness 2.0: staying aware of projects, developers and tasks using dashboards and feeds," in *Proc. ICSE*, pp. 365–374, 2010.

[140] Trompenaars, A. and Hampden-Turner, C., *Riding The Waves of Culture: Understanding Diversity in Global Business*. McGraw Hill, 1998.

[141] ul Haq, S., Raza, M., Zia, A., and Khan, M. N. A., "Issues in global software development: A critical review," *Software Engineering and Applications*, vol. 4, no. 4, pp. 590–595, 2011.

[142] Umarji, M. and Seaman, C., "Conceptualizing software engineering "people " issues," in *Proc. Workshop on Ontology, Conceptualizations and Epistemology for Software and Systems Engineering*, Feb. 07 2005.

[143] Umarji, M. and Seaman, C., "Why do programmers avoid metrics?," in *Proc. Empirical Software Engineering and Management*, (New York, NY, USA), pp. 129–138, ACM, 2008.

[144] van Solingen, R. and Valkema, M., "The Impact of Number of Sites in a Follow the Sun Setting on the Actual and Perceived Working Speed and Accuracy: A Controlled Experiment," in *Proc. ICGSE*, 2010.

[145] Vatrapu, R. K. and Suthers, D. D., "Cultural influences in collaborative information sharing and organization," in *Pro. ICIC*, 2010.

[146] Venolia, G. D., DeLine, R., and LaToza, T., "Software development at microsoft observed," Tech. Rep. MSR-TR-2005-140, Microsoft Research (MSR), Oct. 2005.

[147] Ventoro, "Offshore 2005 Research: Preliminary Findings and Conclusions." `http://www.ventoro.com/Offshore2005ResearchFindings.pdf`, Jan 2005.

[148] Visser, C. and van Solingen, R., "Selecting Locations for Follow-the-Sun Software Development: Towards a Routing Model," in *Proc. ICGSE*, 2009.

[149] Šmite, D., Wohlin, C., Gorschek, T., and Feldt, R., "Empirical evidence in global software engineering: a systematic review," *Empirical Software Engineering*, vol. 15, no. 1, pp. 91–118, 2010.

[150] Walenstein, A., *Cognitive Support in Software Engineering Tools: A Distributed Cognition Framework*. PhD thesis, SIMON FRASER UNIVERSITY, May 2002.

[151] Whittaker, J., "What is software testing? and why is it so hard?SS," *IEEE Software*, vol. 17, pp. 70–79, 2000.

[152] Winkler, J., Dibbern, J., and Heinzl, A., "The impact of cultural differences on outsourced offshoring," *Information Systems Frontiers*, vol. 10, pp. 243–258, 2008.

[153] Yamaura, T., "Why Johnny can't test [software]," *IEEE Software*, vol. 15, no. 2, pp. 113–115, 1998.

[154] Zarrella, E. and Udhas, P., "Strategic evolution: A global survey on sourcing today (KPMG Report)." `http://www.kpmg.com/CN/en/IssuesAndInsights/ArticlesPublications/Documents/global-survey-sourcing-0703.pdf`, March 2007.

[155] Zhang, S. and Shani, D., "Onshore-offshore competition: a stage models," in *Proc. ICIC*, pp. 263–266, 2010.